Psion Teklogix
Mobile Devices SDK

Developers Guide

August 5, 2009 **Part No. 8100016.L**

ISO 9001 Certified
Quality Management System

PSION TEKLOGIX

© Copyright 2009 by Psion Teklogix Inc., Mississauga, Ontario

2100 Meadowvale Boulevard, Mississauga, Ontario, Canada L5N 7J9

http:\\www.psionteklogix.com

This document and the information it contains is the property of Psion Teklogix Inc., is issued in strict confidence, and is not to be reproduced or copied, in whole or in part, except for the sole purpose of promoting the sale of Psion Teklogix manufactured goods and services. Furthermore, this document is not to be used as a basis for design, manufacture, or sub-contract, or in any manner detrimental to the interests of Psion Teklogix Inc.

Windows® and the Windows Logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such trademarks by Psion Teklogix Inc. is under license.

All trademarks are the property of their respective holders.

Return-To-Factory Warranty

Psion Teklogix Inc. provides a return to factory warranty on this product for a period of twelve (12) months in accordance with the Statement of Limited Warranty and Limitation of Liability provided at www.psionteklogix.com/warranty. (If you are not already a member of Teknet and you attempt to view this warranty, you will be asked to register. As a member of Teknet, you will have access to helpful information about your Psion Teklogix products at no charge to you.) In some regions, this warranty may exceed this period. Please contact your local Psion Teklogix office for details. The warranty on Psion Teklogix manufactured equipment does not extend to any product that has been tampered with, altered, or repaired by any person other than an employee of an authorized Psion Teklogix service organization. See Psion Teklogix terms and conditions of sale for full details.

Warning: Psion Teklogix warranties take effect on the date of shipment.

Service

Psion Teklogix provides a complete range of product support services to its customers. For detailed information, please refer to Appendix: “Support Services / Worldwide Offices”. This section also provides information about accessing support services through the Psion Teklogix web site.

Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/EC

This Product, and its accessories, comply with the requirements of the Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/EC. If your end-of-life Psion Teklogix product or accessory carries a label as shown here, please contact your local country representative for details on how to arrange recycling.

For a list of international subsidiaries, please go to:

www.psionteklogix.com/EnvironmentalCompliance

Restriction On Hazardous Substances (RoHS) Directive 2002/95/EC

What is RoHS?

The European Union has mandated that high environmental standards be met in the design and manufacture of electronic and electrical products sold in Europe, to reduce hazardous substances from entering the environment. The "Restriction on Hazardous Substances Directive (RoHS)" prescribes the maximum trace levels of lead, cadmium, mercury, hexavalent chromium, and flame retardants PBB and PBDE that may be contained in a product. Only products meeting these high environmental standards may be "placed on the market" in EU member states after July 1, 2006.

RoHS Logo

Although there is no legal requirement to mark RoHS-compliant products, Psion Teklogix Inc. indicates its compliance with the directive as follows:

The RoHS logo located either on the back of the product or underneath the battery in the battery compartment (or on a related accessory such as the charger or docking station) signifies that the product is RoHS-compliant as per the EU directive. Other than as noted below, a Psion Teklogix product that does not have an accompanying RoHS logo signifies that it was placed on the EU market prior to July 1, 2006, and is thereby exempt from the directive.

Note: Not all accessories or peripherals will have a RoHS logo due to physical space limitations or as a result of their exempt status.

Disclaimer

Every effort has been made to make this material complete, accurate, and up-to-date. In addition, changes are periodically added to the information herein; these changes will be incorporated into new editions of the publication.

Psion Teklogix Inc. reserves the right to make improvements and/or changes in the product(s) and/or the program(s) described in this document without notice, and shall not be responsible for any damages, including but not limited to consequential damages, caused by reliance on the material presented, including but not limited to typographical errors.

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 About This Manual 3
1.2 Text Conventions 4
1.2.1 Command Syntax 5
1.3 Development Environments 5
1.4 Windows Versions 5
1.5 Resetting Psion Teklogix Computers 6
1.5.1 Reset Types and Effects 6
1.5.2 Reset Methods 7
1.6 Non-Psion Teklogix Computers 9
1.7 Other Documentation For The SDK 9

Chapter 2: The C++ Library

2.1 Namespace List 13
2.2 Development Platforms 14
2.3 Getting Started in C++ 14
2.3.1 Selecting the Platform SDKs 14
2.3.2 Including the Psion Teklogix Header Files 14
2.3.3 Adding the Psion TeklogixLibraries 15
2.3.4 Deploying the Psion Teklogix DLL 15
2.4 Troubleshooting 15
2.5 Online Help 16

Chapter 3: The Java Library

3.1 Packages List 19
3.2 Development Platforms 19
3.3 Getting Started in Java 20

Chapter 4: The .NET Library

4.1 Namespaces List 23
4.2 Development Platforms 24
4.3 Getting Started in .NET 24
4.3.1 Importing the Psion Teklogix Toolbox Items 24
4.3.2 Linking to the Mobile Devices SDK Library 24
4.3.3 Add the Common Library Source File 25
4.4 Wizards 25
4.5 Online Help 26

Chapter 5: Backlight

5.1 Backlighting 29
5.1.1 Thresholds 29
5.1.2 Timeouts 29
5.2 Backlight Configuration Parameters 31
5.3 Backlight API Elements 34

Chapter 6: Power Management

6.1 Battery Power Management 37
6.1.1 Suspend Timeout 37
6.1.2 Events 38
6.1.3 Power-on Keyguard 38
6.1.4 Reset Key 38
6.1.5 Battery Information 38
6.1.5.1 Battery Threshold 38
6.1.5.2 Main Battery and Backup Battery Lifetimes And Remaining Charge 39
6.1.6 Power Management API Elements 39
6.2 Card Slots 39
6.2.1 Controlling Power Through The Windows Control Panel 41
6.2.2 Controlling Power Through The SDK 42
6.2.3 Card Slot Control API Elements 43
6.3 Peripherals 43
6.3.1 Power Modes 43
6.3.2 Power Modes API Elements 43

Chapter 7: Display

7.1 Display 47
7.2 Display API Elements 48

Chapter 8: Indicators

8.1 Indicators 51
8.2 Using LED Colours 51
8.3 Controlling Pulses 52
8.4 Controlling Illumination Patterns 52
8.5 Indicator API Elements 52

Chapter 9: Keyboard

9.1 Keyboard 55
9.1.1 Supported Keyboards 55
9.1.2 Disabling The Keyboard 56
9.1.3 Keyboard API Elements 56
9.2 Keyboard Remapping 56
9.2.1 Scan Codes For Psion Teklogix Keyboards 58
9.2.2 Keyboard Remapping API Elements 62
9.3 Key Insertion 62
9.3.1 Key Insertion API Elements 63

Chapter 10: Peripherals

10.1 Overview 67
10.2 Definition Of Terms 67
10.3 Events 67
10.4 Peripheral and Card Slot Power 67
10.5 Docking Station 67
10.6 Tether Ports 68
10.7 Peripheral API Elements 68

Chapter 11: Permanent Storage

11.1 Permanent Storage 71
11.2 Locking Permanent Storage 71
11.3 Permanent Storage API Elements 71

Chapter 12: RAS (Remote Access Service)

12.1 Overview 75
12.2 RAS Architecture 75
12.3 RAS on Windows Mobile (Connection Manager) 76
12.4 RAS API Elements 76

Chapter 13: Remote Sockets

13.1 Overview 79
13.2 802.11 Systems 79
13.3 Narrowband Radio Systems 79
13.3.1 Remote Sockets Proxy 79
13.3.2 Starting A Remote Sockets Connection 79
13.3.3 Ending A Remote Sockets Connection 79
13.3.4 Configuring A Psion Teklogix Network Controller (9400/9500) 80
13.3.5 Configuring A Psion Teklogix Computer 80
13.3.6 Troubleshooting A Remote Sockets Connection 80
13.3.7 Remote Sockets API Elements 82
13.4 Technical References 82

Chapter 14: Scanners

14.1 Overview 85
14.1.1 Types Of Scanners 85
14.1.1.1 Non-decoded Scanners 85
14.1.1.2 Decoded Scanners 85
14.1.1.3 Imaging Scanners 85
14.1.2 Scanner Connections 85
14.1.2.1 Internal Scanners 86
14.1.2.2 External Scanners 86
14.1.3 Symbologies 87
14.2 Configuring Scanners 88
14.2.1 Configuring Through The Windows Control Panel 88
14.2.2 Configuring Through The SDK 89
14.2.3 Configuring With Bar Codes 89
14.3 Configuring Scanners Through The Mobile Devices SDK 89
14.3.1 Configuring Scanner Properties 90
14.3.1.1 Double-click Settings 90
14.3.1.2 Display, Beep, And Logging Options 91
14.3.2 Configuring Global Bar Code Options 92
14.3.2.1 Non-decoded Scanner Options 92
14.3.2.2 Decoded Scanner Options 92
14.3.2.3 Decoded Scanner Advanced Options 93
14.3.2.4 Decoded Scanner 2D Options 93
14.3.2.5 Decoded Scanner Data Options 94
14.3.2.6 Imager Options (Deprecated) 94
14.3.2.7 Imager Advanced Options (Deprecated) 95

14.3.3 Configuring Bar Code Symbologies 96
14.3.3.1 Bar Code Field Settings 96
14.3.3.2 Code 39 Settings 97
14.3.3.3 Trioptic Code Settings 100
14.3.3.4 Code 128 Settings 101
14.3.3.5 EAN 13 Settings 103
14.3.3.6 EAN 8 Settings 105
14.3.3.7 UPC A Settings 108
14.3.3.8 UPC E Settings 110
14.3.3.9 UPC/EAN Shared Settings 113
14.3.3.10 Codabar Settings 114
14.3.3.11 Code 93 Settings 116
14.3.3.12 Code 11 Settings 119
14.3.3.13 Interleaved 2 of 5 Settings 120
14.3.3.14 MSI Plessey Settings 123
14.3.3.15 Matrix 2 of 5 Settings 125
14.3.3.16 Discrete 2 of 5 Settings 126
14.3.3.17 IATA 2 of 5 Settings 128
14.3.3.18 Telepen Settings 129
14.3.3.19 RSS Code Settings 130
14.3.3.20 PosiCode Settings 132
14.3.3.21 Composite Settings 132
14.3.3.22 TLC-39 Settings 134
14.3.3.23 PDF417 Settings 135
14.3.3.24 Micro PDF-417 Settings 137
14.3.3.25 Code 16K Settings 138
14.3.3.26 Code 49 Settings 139
14.3.3.27 Codablock Settings 139
14.3.3.28 2D Data Matrix Settings 140
14.3.3.29 2D QR Code Settings 141
14.3.3.30 2D MaxiCode Settings 142
14.3.3.31 2D Aztec Settings 143
14.3.3.32 Postal: PlaNET Settings 144
14.3.3.33 Postal: PostNET Settings 145
14.3.3.34 Postal: Australian Settings 146
14.3.3.35 Postal: Canadian Settings 146
14.3.3.36 Postal: Japanese Settings 147
14.3.3.37 Postal: Kix Settings 148
14.3.3.38 Postal: Korean Settings 149
14.3.3.39 Postal: Royal Settings 150
14.4 Reading Bar Codes 150
14.4.1 Initiating A Bar Code Scan 150
14.4.2 Controlling A Bar Code Scan 151
14.5 Scanner Events 152

Chapter 15: Serial Ports

15.1 Overview 155
15.2 WORKABOUT PRO G2 Serial Port Assignments 155
15.3 7530, 7535, 8525, and 8530 Serial Port Assignments 156
15.4 8515 Serial Port Assignments 156
15.5 Ikôn Serial Port Assignments 157
15.6 **NEO** Serial Port Assignments 157

15.7 Events 157
15.8 Java 158
15.9 Serial Port API Elements 158

Chapter 16: Sound

16.1 Sound Hardware 161
16.2 Playing Beeps 161
16.3 Playing WAV Audio Format Files 161
16.4 Sound API Elements 162

Chapter 17: System Information

17.1 System Information 165
17.2 Machine Type 165
17.3 Model 166
17.4 Unique Machine Identifier 166
17.5 Psion Teklogix Build Codes 166
17.6 Setting The Ratio Of Program Memory To Storage Memory 166
17.7 System Information API Elements 167

Chapter 18: Total Recall

18.1 Total Recall 171
18.2 Cold Resets And Clean Starts 171
18.3 Backup Profile 171
18.4 Profile Types 171
18.5 Profile Content 171
18.6 Storage Locations For Profiles 172
18.7 Restoring Profiles 172
18.8 Troubleshooting 173
18.9 Total Recall API Elements 173

Chapter 19: Windows Shell

19.1 Windows Shell 177
19.2 Setting Windows Security 177
19.3 Enabling And Disabling The Windows Shell 177
19.4 Security Level Change Event 177
19.5 System Security API Elements 178

Chapter 20: Trigger Control

20.1 Overview 181
20.2 Definition Of Terms 181
20.3 Trigger Consumer Registration 181
20.4 Trigger Source IDs 181
20.4.1 Virtual Key Codes 182
20.5 Trigger Associations 182
20.5.1 Trigger Control Flags 182
20.6 Double-Clicks 183
20.7 Events 183
20.7.1 Simulated Events 184

20.7.2 Single-Click Events And Double-Click Events. 184
20.8 Trigger Control API Elements. 184

Chapter 21: Wireless Local-Area Networking

21.1 Wireless Local-Area Networking (WLAN). 187
21.1.1 Radios Supported. 187
21.1.2 Configuring WLAN Through The Control Panel. 187
21.1.3 Wireless Zero Config 187
21.1.4 Summit Client Utility. 187
21.1.5 Odyssey Access Client by Juniper Networks. 187
21.2 Availability Of API Libraries. 188
21.3 Encryption For Data Transmission 188
21.3.1 Wired Equivalent Privacy (WEP) Keys 188
21.4 Connection State Parameters 188
21.4.1 Received Signal Strength Indicator—RSSI 188
21.4.2 Transmit Power. 188
21.4.3 Network adaptor Name 189
21.5 Summit Radio Features 189
21.5.1 Summit Mode 189
21.5.2 Wireless Zero Config Mode (WZC) 189
21.5.3 Changing Between Modes 189
21.6 WLAN API Elements. 189

Chapter 22: Wireless Wide-Area Networking

22.1 Wireless Wide-Area Networking (WWAN) 193
22.2 Psion Teklogix Computers Supporting WWAN 193
22.3 Supported WWAN Modems 193
22.3.1 Multiplexing. 194
22.3.2 Virtual Serial Port 194
22.3.2.1 Using The Virtual Serial Port 194
22.3.2.2 Configuration 194
22.3.2.3 AT Commands 195
22.3.2.4 Entering AT Commands in Windows CE 195
22.3.3 Dial-up Data Connections 196
22.3.3.1 Dial-up Data in GSM Networks 196
22.3.3.2 Dial-up Data in UMTS Networks. 197
22.3.3.3 Dial-up Data in CDMA Networks 197
22.3.3.4 Dial-up Data in iDEN Networks. 197
22.3.4 Packet Data. 197
22.3.5 Summary Of Modem Differences. 198
22.3.6 GSM Power Driver 198
22.4 Initializing WWAN 199
22.4.1 Checking The Initialization Status Of The WWAN Driver 199
22.4.2 WWAN Driver Status Flags 199
22.4.3 Initializing The WWAN Driver 201
22.5 Closing WWAN 205
22.6 Connecting To The Internet 205
22.6.1 Windows CE 5.0 Connection 206
22.6.2 Windows Mobile 5.0 And Windows Mobile 6 Connection. 209
22.6.3 The DbGprs.csv File. 209

22.6.4 Checking Packet Data Status 210
22.6.5 Roaming 210
22.6.6 Connecting Manually to A WWAN Network 210
22.6.7 Selecting An Access Point Name (APN) 211
22.6.8 Connecting To A VPN 211
22.7 Access Flags 212
22.7.1 Controlling The Interaction With The Control Panel 213
22.8 RSSI (Received Signal Strength Indicator) 213
22.9 WWAN API Elements 214
22.10 Using SMS (Short Message Service) 215
22.10.1 SMS API Elements 215
22.11 WWAN Supplementary Services 216
22.11.1 Voice Service States 217
22.12 Voice Calls On A WWAN 217
22.12.1 Initializing WWAN 217
22.12.2 Initializing Voice Over WWAN 217
22.12.3 Closing Voice Over WWAN 220
22.12.4 Making A Phone Call 220
22.12.5 Voice Call States 222
22.12.6 Receiving A Phone Call 224
22.12.7 Processing Voice Calls 225
22.12.8 Terminating Voice Calls 227
22.12.9 Call Forwarding 227
22.12.10 Dual-Tone Multifrequency (DTMF) 228
22.12.11 Blocking Inbound And Outbound Calls 228
22.12.12 Blocking Caller ID On Outgoing Calls 228
22.12.13 Audio for Voice Over WWAN 228
22.12.14 Voice Over WWAN Events 229
22.12.14.1 Voice Call Status Changed Event 229
22.12.14.2 Voice Call Manager Incoming Call Event 230
22.12.15 Voice Over WWAN API Elements 230
22.13 Phone Books 230
22.13.1 Data Coding Schemes For Phone Book Entries 231
22.13.2 Phone Book Types 232
22.13.3 Phone Book API Elements 232
22.14 Resource Materials 233

Appendix A: Resources

A.1 Manuals and URLs A-1

Appendix B: Registry Keys

B.1 WORKABOUT PRO Registry Keys B-1
B.1.1 Registry Settings For Controlling VGA B-1
B.1.2 Registry Settings For Controlling Scanner Power B-1
B.1.3 Registry Settings For Serial Ports B-1
B.1.4 Registry Settings For Psion Teklogix Device Drivers B-2
B.1.5 Registry Settings For Non-Psion Teklogix Device Drivers B-2
B.1.5.1 Loading Non-Psion Teklogix Drivers B-3

Index I

1 INTRODUCTION

1.1 About This Manual 3
1.2 Text Conventions 4
1.2.1 Command Syntax. 5
1.3 Development Environments 5
1.4 Windows Versions 5
1.5 Resetting Psion Teklogix Computers. 6
1.5.1 Reset Types and Effects 6
1.5.2 Reset Methods 7
1.6 Non-Psion Teklogix Computers 9
1.7 Other Documentation For The SDK 9

1.1 About This Manual

This manual provides guidance on creating applications for Psion Teklogix devices running Microsoft Windows® CE or Windows Mobile operating systems, using the Mobile Devices SDK. The information in this manual is current with version 3.1 of the SDK software.

Chapter 1: Introduction

provides an overview of this manual, the other documentation available with the SDK, and text conventions used.

Chapter 2: The C++ Library

describes how to use the SDK's C++ namespaces.

Chapter 3: The Java Library

describes how to use the SDK's Java packages.

Chapter 4: The .NET Library

describes how to use the SDK's .NET assembly, which allows developers to program in the .NET Compact Framework, using languages such as Visual Basic.Net or C#.

Chapter 5: Backlight

describes how to control the backlights on Psion Teklogix computers.

Chapter 6: Power Management

describes how to control battery and external power supplies.

Chapter 7: Display

describes the processes for obtaining display screen information.

Chapter 8: Indicators

describes how to manipulate the LEDs.

Chapter 9: Keyboard

describes how to disable the keyboard and how to remap scan codes.

Chapter 10: Peripherals

describes how to detect and control tether ports and docking stations.

Chapter 11: Permanent Storage

describes how to access and use permanent storage.

Chapter 12: RAS (Remote Access Service)

describes how to use the Remote Access Service (RAS).

Chapter 13: Remote Sockets

describes how to use remote sockets in C++ applications.

Chapter 14: Scanners

describes the configuration of scanners and bar code symbologies.

Chapter 15: Serial Ports

describes how to detect serial ports and serial port change events.

Chapter 16: Sound

describes how to control the beeper and how to play WAV files.

Chapter 17: System Information

describes how to control the Windows security and how to obtain hardware and software information.

Chapter 18: Total Recall

describes how to control the Total Recall utility programmatically.

Chapter 19: Windows Shell

describes how to set security levels and control access to the Windows shell.

Chapter 20: Trigger Control

describes how to control the trigger sources on a Psion Teklogix computer.

Chapter 21: Wireless Local-Area Networking

describes how to implement WLAN.

Chapter 22: Wireless Wide-Area Networking

describes how to implement WWAN.

Appendix A: Resources

lists other documents and web sites where you can find information related to developing with the Mobile Devices SDK.

Appendix B: Registry Keys

lists and describes certain WORKABOUT PRO registry keys which may be useful when developing with the Mobile Devices SDK.

1.2 Text Conventions

The following conventions and syntax are followed throughout this document, with the exception of when referencing API commands (see Section 1.2.1: “Command Syntax”):

- Instructions to press specific keys on the keypad are indicated with the name or symbol of the key between square brackets.
  e.g. [SPACE], [TAB], [BLUE], [A], [.], etc.
- Instructions to press buttons with dedicated functions are given with the name or function of the button in bold type.
  e.g. **Power**, **Scan**, etc.
- Instructions to type a specific string of text are given between quotation marks.
  e.g. Type “exit”, and press [ENTER].

Note: Notes highlight additional helpful information.

Important: These statements provide important instructions or additional information that is critical to the operation of the computer or other equipment.

Warning: These statements provide important information that may prevent injury, damage to the equipment, or loss of data.

An arrow next to field description information (usually in tables) indicates a recommended or suggested configuration setting.

1.2.1 Command Syntax

When commands are described in text the following conventions are used in the manual:

- Elements that must be typed exactly as shown in the text are in **bold**.
- Elements that are placeholders are in *italic*.

The general form of a command is as follows:

sample {**+r** | **-r**} *argument* ... [*option*]

Where:

Element	Meaning
sample	Indicates the name of the command or utility.
{ }	Surrounds a set of choices from which you must choose one.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices, not the symbol.
argument	Specifies a variable name or other information that you must provide, such as a path and file name.
...	Indicates that you can type multiple arguments of the same type. Type only the information, not the ellipsis (...).
[ ]	Indicates one or more optional items. Type only the information within the brackets, not the brackets themselves.

1.3 Development Environments

As of this printing, the following development environments can be used to develop applications with the Mobile Devices SDK:

- Microsoft® Visual Studio 2005
- Microsoft® Visual Studio 2008

To develop applications for Windows Mobile devices in the Visual Studio environment, you should also install the Windows Mobile 6 SDKs available from Microsoft:

http://www.microsoft.com/downloads/details.aspx?familyid=06111A3A-A651-4745-88EF-3D48091A390B&displaylang=en

1.4 Windows Versions

The Mobile Devices SDK is used to develop applications for the following Windows operating systems:

- Windows Embedded CE 5.0
- Windows Mobile 5.0
- Windows Mobile 6 Classic
- Windows Mobile 6 Professional

Table 1.1 shows the availability of the different operating systems on the various Psion Teklogix computer devices.

Table 1.1 Availability Of Windows Operating Systems

	Windows Embedded CE 5.0	Windows Mobile 5.0	Windows Mobile 6 Classic & Professional
7530 G2	Yes	No	No
7535 G2	Yes	No	No
8515	Yes	No	No
8525 G2	Yes	No	No
8530 G2	Yes	No	No
Ikôn	Yes	No	Yes
NEO	Yes	No	No
WORKABOUT PRO G1	No	Yes	No
WORKABOUT PRO G2	Yes	No	Yes

1.5 Resetting Psion Teklogix Computers

This section describes the different methods and effects of resetting Psion Teklogix computers.

1.5.1 Reset Types and Effects

Suspend

The most common form of computer reset is the **suspend** state, which may occur automatically after a period of inactivity, or when the battery power reserve drops below a specified threshold, and is also typically what occurs when the user manually powers down the computer.

In suspend mode, a computer's display, keyboard and peripherals are powered down. The operating system and any running applications are maintained in their current state, until the power source depletes. The user can "wake" or "resume" the computer by powering it on again. This process is known as a "**suspend/resume cycle**".

Warm Reset and Cold Reset

There are two primary types of reset available on a Psion Teklogix computer, referred to as **warm reset** and **cold reset** procedures. The specific actions taken during the reset depend on the operating system, but in general the warm reset is a simple reboot of the operating system and the cold reset is a complete reset of the device at the hardware level.

Windows CE Operating Systems

When you perform a warm reset on a Psion Teklogix computer running Windows CE, the operating system restarts **without** clearing the object store memory where the file system resides. Registry settings, installed programs and any data files are preserved. Any open applications will be closed and any unsaved data will be lost.

A cold reset clears the object store memory and restarts the operating system. Any registry setting changes are maintained, but all installed programs and data files not located on permanent memory storage media such as the "Flash Disk" partition or an SD card, will be lost.

Windows Mobile Operating Systems

On a Psion Teklogix computer running Windows Mobile, a warm reset completely reloads and re-starts the operating system. Registry settings, installed programs, and any data files that reside in flash memory are all preserved through the reset.

Files and data stored in the RAM Drive partition are **not** preserved during a warm reset. To have files stored in the RAM Drive persist through the reset procedure, you must create a Total Recall profile (see Chapter 18: "Total Recall") and store it on the flash ("My Device") file system.

A cold reset in Windows Mobile is identical to a warm reset, except some low level hardware (PCon) is also restarted before the operating system boots.

Important: ***To initiate a computer reset from within a program on a computer running Windows Mobile, you must call the warm reset method. Attempting to call the cold reset method on a Windows Mobile device will fail.***

Clean Start

In addition to the standard reset functions, there is a **clean start** procedure that resets the computer to its factory default settings. This operation can only be performed manually, and cannot be initiated through a program call. This should only be used when absolutely necessary to recover the computer to a usable state as it involves booting to the bootstrap (BooSt) menu, which should typically only be accessed by trained Psion Teklogix personnel. ***Use with caution!***

Table 1.2 summarizes the effects on the computer's memory and settings for each type of reset.

Table 1.2 Effects of Resetting Psion Teklogix Computers

Feature	Warm Reset	Cold Reset	Clean Start
Restarts Windows	Yes	Yes	Yes
Restarts all drivers	Yes	Yes	Yes
Clears application memory (RAM)	Yes	Yes	Yes
Sets object store RAM to factory default	No	Yes	Yes
Sets registry to factory default	No	No	Yes
Sets image to factory default	No	No	Yes
Clears addressable persistent storage (flash)	No	No	No
Can be invoked with API function call	Yes	Yes (CE) No (WM)	No

1.5.2 Reset Methods

There are several methods available for invoking the different reset types on Psion Teklogix computers.

The suspend, warm reset and cold reset types can all be initiated from within a program via the Mobile Devices SDK. These reset types also appear as user-selectable options in the Windows *Shutdown* menu on computers running the Windows CE 5.0 operating system.

The clean start reset can only be performed manually by the computer user.

Table 1.3 lists how to execute all the different reset types manually on Psion Teklogix computers using the keypad.

Table 1.3 Methods for Resetting Psion Teklogix Computers

	Suspend	Warm Reset	Cold Reset	Clean Start
753x	Press [BLUE], then press [ENTER].	Press and hold [BLUE] and [ENTER] for six seconds.	Press and hold **Scan**, [BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold **Scan**, [BLUE] and [ENTER] for six seconds. At BooSt menu, press [!].
8515	Press [BLUE], then press [ENTER].	Press and hold [BLUE] and [ENTER] for six seconds.	Press and hold [SPACE], [BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold [SPACE], [BLUE] and [ENTER] for six seconds. At BooSt menu, press [!].
8525 / 8530	Press [BLUE], then press [ENTER].	Press and hold [BLUE] and [ENTER] for six seconds.	Press and hold [SPACE], [BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold [SPACE], [BLUE] and [ENTER] for six seconds. At BooSt menu, press [!].
Ikôn (Windows CE)	Press **Power** button.	Press and hold [BLUE] and [ENTER] for six seconds.	Press and hold **Left Scan**, [BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold **Left Scan**, [BLUE] and [ENTER] for six seconds. At BooSt menu, press [!].
Ikôn (Windows Mobile)	Press **Power** button.	Press and hold [BLUE] and [ENTER] for six seconds.	Press and hold **Left Scan**, [BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold **Left Scan**, [BLUE] and [ENTER] for six seconds. At BooSt menu, press [.], then type either “clean” (alpha keypad) or “25326” (numeric keypad), followed by [ENTER].
NEO	Press [FN/BLUE], then press [ENTER].	Press and hold [FN/BLUE] and [ENTER] for six seconds.	Press and hold **Scan**, [FN/BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold **Scan**, [FN/BLUE] and [ENTER] for six seconds. At BooSt menu, press [!].

Table 1.3 Methods for Resetting Psion Teklogix Computers

	Suspend	Warm Reset	Cold Reset	Clean Start
WORKABOUT PRO (Windows CE)	Press [FN/BLUE], then press [ENTER].	Press and hold [FN/BLUE] and [ENTER] for six seconds.	Press and hold **Scan**, [FN/BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold **Scan**, [FN/BLUE] and [ENTER] for six seconds. At BooSt menu, press [!].
WORKABOUT PRO (Windows Mobile)	Press [FN/BLUE], then press [ENTER].	Press and hold [FN/BLUE] and [ENTER] for six seconds.	Press and hold **Scan**, [FN/BLUE] and [ENTER] for six seconds. At BooSt menu, press [1].	Press and hold **Scan**, [FN/BLUE] and [ENTER] for six seconds. At BooSt menu, press [.], then type “25326”, followed by [ENTER].

1.6 Non-Psion Teklogix Computers

The Mobile Devices SDK cannot be used on computers other than those made by Psion Teklogix. Attempts to load a Psion Teklogix DLL file on a non-Psion Teklogix computer will fail with an error message.

1.7 Other Documentation For The SDK

There are three categories of manuals that should be used when programming with the Mobile Devices SDK.

This manual, the *Developers Guide*, provides an overview of the SDK and provides information on how to use the individual language libraries. A single *Developers Guide* covers all libraries and devices supported by the SDK. This manual is available in Portable Document Format (PDF).

API Reference Manuals are provided for each language library, in a format appropriate to that language. All information specific to the language libraries, including class, method, field, and property descriptions, are captured by the *API Reference Manuals*.

In addition to the developer and API documentation, each Psion Teklogix computer has a dedicated *User Manual*. While the SDK does support emulators, you will need to obtain an actual device in order to test device-specific features such as bar code scanners. The *User Manuals* provide valuable help in getting acquainted with the features of these devices.

THE C++ LIBRARY 2

2.1 Namespace List 13
2.2 Development Platforms 14
2.3 Getting Started in C++ 14
2.3.1 Selecting the Platform SDKs 14
2.3.2 Including the Psion Teklogix Header Files 14
2.3.3 Adding the Psion TeklogixLibraries 15
2.3.4 Deploying the Psion Teklogix DLL 15
2.4 Troubleshooting 15
2.5 Online Help 16

2.1 Namespace List

The Psion Teklogix Mobile Devices SDK for C++ contains the following namespaces within the **PsionTeklogix** namespace:

Namespace	Description
Backlight	Configures and queries the backlight.
DisplayInformation	Returns information on the display hardware.
Indicators	Controls the LEDs.
Keyboard	Gives access to the keyboard, including keyboard remapping.
Keyboard::NoChording	Deprecated — use KeyRemapper class in Keyboard namespace instead.
Orientation	Detects and controls the orientation of the display.
PowerManagement	Gives access to the battery and external power states and the associated events.
RemoteSockets	Enables applications to communicate with each other through a BSD-like socket interface. Only available for Psion Teklogix computers with a Narrow Band radio.
RFID	Enables configuration and control of RFID tag readers. Deprecated — use RFID SDK instead.
Scanner	Controls and reads data from bar code readers and imagers.
SerialPorts	Enables the detection of serial ports on Psion Teklogix computers. It also provides information on these serial ports, and listeners for physical change events that arise on these ports.
Sound	Provides control of the beeper and/or speaker.
SystemPTX	Contains device-specific hardware and software information.
System::Peripherals	Enables the configuration and manipulation of peripheral devices such as docking stations, tether ports, and card slots.
System::Peripherals::PowerMode	Contains constants required by the System::Peripherals namespace.
System::Security	Detects changes to, returns, and configures the security settings of the computer.
System::SystemInformation	Provides information on the hardware and software of the Psion Teklogix computer.
TotalRecall	Enables the programmatic control of the Total Recall utility.

Namespace	Description
Trigger	Controls the pistol grip trigger and the [SCAN] button, and enables keyboard keys to be used as triggers.
Utility	Contains structures and typedefs that are used in several other namespaces.
WLAN	Enables wireless local-area networking.
WWAN	Enables wireless wide-area networking.

2.2 Development Platforms

The following development platforms should be used for developing C++ applications with the Mobile Devices SDK:

- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

2.3 Getting Started in C++

The following simple steps describe what is required to begin developing an application using the Mobile Devices SDK in Visual Studio 2005 or 2008. These are only the steps specific to incorporating the SDK; it is assumed the user has familiarity with the Visual Studio environment.

In addition, sample C++ projects can be found in the \Samples\cpp subdirectory of your Mobile Devices SDK installation.

2.3.1 Selecting the Platform SDKs

Platform SDKs contain certain platform files such as <windows.h> which are required for developing C++ applications.

1. When the Application Wizard appears after specifying the application type and name, select **Platforms** from the items on the left.
2. Make sure that the *Selected SDKs* box contains the appropriate SDKs for the platform(s) you are developing for (**PsionTeklogixCE500, Windows Mobile 6**).

Note: For information on installing the Windows Mobile SDK, see Section 1.3: "Development Environments".

2.3.2 Including the Psion Teklogix Header Files

1. In the opening #include statements of your main program, add the following:
   #include <PsionTeklogix.hpp>
2. Open the *Project* menu and select **[Project Name] Properties...**.
3. Select **Configuration Properties > C/C++ > General** from the tree structure on the left.
4. Click in the field next to **Additional Include Directories** and add the following path (if you installed the SDK to a directory other than the default, change the path accordingly):
   C:\Program Files\Psion Teklogix\Mobile Devices SDK V3.1\Cpp
5. Click **OK**.

2.3.3 Adding the Psion TeklogixLibraries

1. Open the *Project* menu and select **[Project Name] Properties...**.
2. Select **Configuration Properties > Linker > Input** from the tree structure on the left.
3. Click in the field next to **Additional Dependencies** and add the following path:
   C:\Program Files\Psion Teklogix[1]\Mobile Devices SDK V3.1\<OS Directory>[2]\<VS Directory>[3]\PtxSdkCpp.lib

   [1] If you installed the SDK to a directory other than the default, change the path accordingly.

   [2] Replace with the subdirectory appropriate to the OS you will be deploying your application (**Windows CE 5.00**, **WindowsMobile5**, **WindowsMobile6**).

   [3] Replace with the subdirectory that corresponds to your version of Visual Studio (**Studio2005**, **Studio2008**).

2.3.4 Deploying the Psion Teklogix DLL

When you deploy your application to the device, the files **PtxSdkCommon.dll** and **PtxSdkCpp.dll** must be deployed as well. These files can be found in the same directory as the **PtxSdkCpp.lib** file in the previous step. Copy the files to either the \Windows directory, or to the directory where the rest of the project files reside. The .dll files will then be included automatically when you deploy the solution in Visual Studio.

2.4 Troubleshooting

1. An error message may be generated during deployment because the code is compiled on a platform with a different name. If you see this message, simply click **OK**. This does not have any effect at run time.
2. Fatal Error C1189 may appear when you attempt to build a Debug version of the project. To prevent this error either build a Release version instead, or do the following:
   i. Open the *Project* menu and select **[Project Name] Properties...**.
   ii. Select **Configuration Properties > C/C++ > Code Generation** from the tree structure on the left.
   iii. Click in the field next to *Runtime Library* and change the selection to **Multi-threaded (/MT)**.
   iv. Select **Configuration Properties > C/C++ > Preprocessor** from the tree structure on the left.
   v. Click in the field next to *Preprocessor Definitions* and delete the text **_DEBUG;** from the string.
   vi. Click **OK**.

2.5 Online Help

Online help files for the C++ APIs can be accessed from within the Visual Studios environment. To access these help files, they must be registered with the Visual Studio program.

1. From the directory where you installed the Mobile Devices SDK, go to the **documentation** subdirectory. Extract the files contained in **Mobile Devices SDK CPP.zip** to a separate directory on the computer.
2. Download and install Helpware's *H2Reg* shareware program available from http://helpware.net/mshelp2/h2reg.htm.
3. Copy the files **H2Reg.exe** and **H2Reg.ini** from the H2Reg installation directory into the same directory with the extracted help files.
4. Open a Command Prompt window and navigate to the directory with the extracted and downloaded files.
5. From the Command Prompt run the following command:
H2Reg.exe -R "cmdfile=H2Reg_cmd.ini".

THE JAVA LIBRARY 3

3.1 Packages List 19
3.2 Development Platforms 19
3.3 Getting Started in Java 20

3.1 Packages List

The Psion Teklogix Mobile Devices SDK contains the following packages within the com.teklogix package:

Package	Description
backlight	Controls the display and keyboard backlights on Psion Teklogix computers.
configuration	Contains controls for accessing and changing registry keys.
display	Returns information on the display hardware.
indicators	Controls the software accessible LEDs on a Psion Teklogix computer.
keyboard	Gives access to the keyboard, including keyboard remapping.
keyboard.FunctionData	Defines system reset function keys.
keyboard::NoChording	Deprecated — use keyboard remapping in keyboard package instead.
power	Gives access to the battery and external power states and the associated events.
ras	Supports network connections over a wide area network—WAN—or over a virtual private network—VPN.
scanner	Configures and controls bar code scanners and imagers.
serial	Enables the detection of serial ports on Psion Teklogix computers and control of the transmission of data through the serial ports.
sound	Enables the playing of beeps and WAV files.
system	Enables the obtaining of hardware information, and the changing of security settings.
totalrecall	Enables the programmatic control of the Total Recall utility.
trigger	Controls the pistol grip trigger and the [SCAN] button, and enables keyboard keys to be used as triggers.
util	Contains classes and packages that are used by several other packages.
util.nativeWindows	Provides access to native Windows events.
WLAN	Enables wireless local-area networking.

3.2 Development Platforms

The Mobile Devices SDK is compatible with Java 2 Software Development Kit (J2SDK) version 1.2.2, and the IBM J9 version of Java Virtual Machine.

The J2SDK is available for download at http://java.sun.com/products/archive/.

The IBM J9 JVM is available as part of the IBM WebSphere Everyplace Micro Environment, which can be found at http://www-01.ibm.com/software/wireless/weme/.

3.3 Getting Started in Java

The Psion Teklogix APIs for Java are contained in the **PsionTeklogixJava.jar** file, located in the \Java subdirectory of your Mobile Devices SDK installation. Include this file when you compile your Java programs using the Java Development Kit. You must also include the file **PtxSdkCommon.dll**, which can be located in any of the **<OS Directory>\<VS Directory>** sub-directories of your Mobile Devices SDK installation (e.g. "**...\Windows CE 5.00\Studio2005**").

Sample Java programs can be found in the \Samples\java subdirectory of your Mobile Devices SDK installation.

THE .NET LIBRARY 4

4.1 Namespaces List 23
4.2 Development Platforms 24
4.3 Getting Started in .NET 24
4.3.1 Importing the Psion Teklogix Toolbox Items 24
4.3.2 Linking to the Mobile Devices SDK Library 24
4.3.3 Add the Common Library Source File 25
4.4 Wizards 25
4.5 Online Help 26

4.1 Namespaces List

The Mobile Devices SDK for .NET has the following namespaces in the PsionTeklogixNet.dll file:

Namespace	Description
PsionTeklogix	Contains all the other namespaces.
Backlight	Controls the display and keyboard backlights on Psion Teklogix computers.
Barcode	Controls and reads data from bar code readers and imagers.
Barcode.ScannerServices	Controls imaging scanners and gives access to data read by them.
Configuration	Contains controls for accessing and changing registry keys.
Indicators	Contains everything needed to control the software accessible LEDs on a Psion Teklogix computer.
Keyboard	Gives access to the keyboard, including keyboard remapping.
Keyboard.NoChording	Deprecated — use KeyRemapper class in Keyboard namespace instead.
Peripherals	Enables the configuration and manipulation of peripheral devices.
Power	Gives access to the battery and external power states and the associated events.
RAS	Supports network connections over a wide area network—WAN— or over a virtual private network—VPN.
Resources	Gives access to the resource files (.res files) on a Psion Teklogix computer.
RFID	Controls and accepts data from RFID tag readers.
Serial	Enables the detection of serial ports on Psion Teklogix computers and control of the transmission of data through the serial ports.
Sound	Enables the playing of beeps and WAV files.
Sound.Beeper	Enables the playing of beeps or tones on the beeper or the speaker.
SystemPTX	Enables the obtaining of hardware information, and the changing of security settings.
TotalRecall	Enables the programmatic control of the Total Recall utility.
Trigger	Contains all the API elements for controlling the pistol grip trigger and the [SCAN] button, and for using keyboard keys as triggers.
Utility	Contains classes that are used in several other namespaces.

Namespace	Description
Windows.Forms	Enables the processing of screen-captured signatures, and access to some of the properties of the Psion Teklogix computer's display.
WLAN	Enables wireless local-area networking.
WWAN	Enables wireless wide-area networking.

4.2 Development Platforms

The following development platforms should be used for developing .NET applications with the Mobile Devices SDK:

- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

Important: ***At this time the Mobile Devices SDK is compatible with Microsoft .NET Compact Framework version 2.0 only. Developing applications using later versions of the .NET CF may prevent access to certain resources or result in compilation failure.***

4.3 Getting Started in .NET

The following simple steps describe what is required to begin developing a .NET application using the Mobile Devices SDK in Visual Studio 2005 or 2008. These are only the steps specific to incorporating the SDK; it is assumed the user has familiarity with the Visual Studio environment.

In addition, sample .NET projects can be found in the \Samples\dotnet subdirectory of your Mobile Devices SDK installation.

4.3.1 Importing the Psion Teklogix Toolbox Items

Note: This step only needs to be done once before you begin developing your first application. Once the toolbox items are imported they will remain in the Visual Studio toolbox until you remove them.

1. In the *Tools* menu select **Choose Toolbox Items...**.
2. In the *.NET Framework Components* tab, click **Browse**.
3. Browse to the ***Mobile Devices SDK* V3.1\Dotnet2** subdirectory.
4. Select the *PsionTeklogixNet.dll* file and click **Open**. The list of components should now include items from the PsionTeklogix namespace.
5. Click **OK** to close the dialog box.

4.3.2 Linking to the Mobile Devices SDK Library

1. After creating a new project, open the *Project* menu and select **Add Reference...**.
2. In the *Browse* tab, use the navigation tools to browse to the ***Mobile Devices SDK* V3.1\DotNet2** subdirectory.
3. Select the *PsionTeklogixNet.dll* file and click **OK**. The PsionTeklogixNet object should now appear in the *Solution Explorer* window under *References*.

4.3.3 Add the Common Library Source File

1. Open the *Project > Add Existing Item...* menu and select **Add Existing Item...**.
2. In the dialog box that appears, use the navigation tools to browse to the *Mobile Devices SDK V3.1* installation directory, and from there to the subdirectory that corresponds to the Operating System the application will be deployed on (**Windows CE 5.00**, **WindowsMobile5**, or **WindowsMobile6**).
3. Browse to the subdirectory that corresponds to your development environment (*Studio2005* or *Studio2008*).
4. Change the selection in the *Objects of type:* field to *Executable Files (*.exe;*.dll;*.ocx)* to make the DLL files viewable.
5. Select the **PtxSdkCommon.dll** file and click **Add**. The DLL file should now appear in the *Solution Explorer* window under the project name.
6. In the *Solution Explorer* window, right-click the *PtxSdkCommon.dll* object and select **Properties** from the drop-down menu that appears.
7. In the *Copy to Output Directory* field of the *Properties* window, select **Copy if newer**.

You should now be ready to begin developing and deploying applications for your Psion Teklogix computers.

4.4 Wizards

The Mobile Devices SDK includes an application wizard which will generate sample .NET programs to demonstrate simple application functions and provide a jumping-off point for creating your own more fully-featured applications.

Note: The wizard feature is only available in Visual Studio 2008.

Follow these steps to access the wizard:

1. Start a new project in Visual Studio.
2. In the *Project Types* frame, select **Visual C# > Smart Device**.
3. In the *Templates* window, select **Psion Teklogix Device Project**.
4. Fill in the project/solution name in the appropriate fields and click **OK** to start the application wizard.
5. If given the option to *Select an SDK*, select **Mobile Devices**.
6. In the *Select a Wizard* box, select one of the following:
   - **None** - creates a simple “Hello World” program with a textbox and button.
   - **Scanner** - creates a simple bar code scanner application to initiate a scan and display the result.
   - **WWAN Phonebook** - creates a simple phone book application that reads and displays the entries on a device’s SIM card.
   - **WWAN Voice** - creates a simple phone application that can initiate and receive voice calls.

4.5 Online Help

Online help files for the .NET APIs can be accessed from within the Visual Studios environment. To access these help files, they must be registered with the Visual Studio program.

1. From the directory where you installed the Mobile Devices SDK, go to the *documentation* subdirectory. Extract the files contained in **Mobile Devices SDK NET.zip** to a separate directory on the computer.
2. Download and install Helpware's **H2Reg** shareware program available from http://helpware.net/mshelp2/h2reg.htm.
3. Copy the files **H2Reg.exe** and **H2Reg.ini** from the H2Reg installation directory into the same directory with the extracted help files.
4. Open a Command Prompt window and navigate to the directory with the extracted and downloaded files.
5. From the Command Prompt run the following command:
H2Reg.exe -R "cmdfile=H2Reg_cmd.ini".

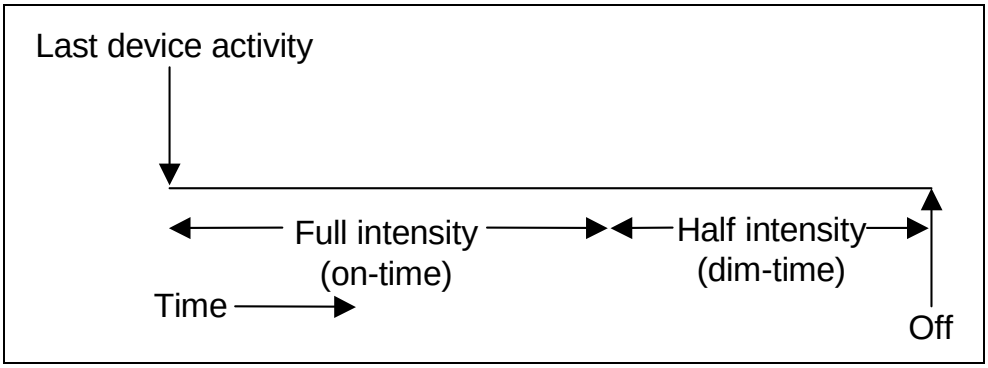
BACKLIGHT 5

5.1 Backlighting 29
5.1.1 Thresholds 29
5.1.2 Timeouts 29
5.2 Backlight Configuration Parameters 31
5.3 Backlight API Elements 34

5.1 Backlighting

The Mobile Devices SDK provides functions that control the display and keyboard backlights on Psion Teklogix computers. The intensity of the backlight and the conditions under which it is activated can be queried and set using the SDK.

To conserve battery power, you can configure the backlights to switch off or dim to half intensity after the computer has been inactive for a selected length of time. The following diagram shows how these times are related:



The last device activity is one of the following:

- A key is pressed on the keyboard.
- The touchscreen is pressed.
- The scanner trigger is pressed.
- Data is received from the host.

If the computer is operating with external power, you can configure the backlights to remain on at all times.

5.1.1 Thresholds

On Psion Teklogix computers with ambient light sensors (7530 & 7535), there is a threshold value that specifies the light level at which the backlights will turn on. The values for the display and keyboard backlight thresholds are configured independently.

Threshold values are integers between 0 (zero) and 100. A value of 0 ensures that the selected backlight is always off. A value of 100 ensures that the backlight can turn on at all lighting levels. Intermediate values control the level of ambient light at which the backlight turns on. The lower the value, the darker it must be before the keyboard backlight can turn on. Regardless of the threshold settings, the backlights will only come on if there is activity to trigger it, such as a keyboard or touch screen event.

5.1.2 Timeouts

Using the Mobile Devices SDK, the backlight on-times and dim-times can be set to any positive integer within the range of the parameter (typically 0 to 2147483647, measured in milliseconds). However, when setting the values using the Control Panel, the choices are limited to a handful of predetermined values presented in a drop-down list.

For all Psion Teklogix computers that have backlights, on-time and dim-time values are available as follows in the Control Panel:

	Windows CE			Windows Mobile	
Time (milliseconds)	Display Backlight On-time	Display Backlight Dim-time	Keyboard Backlight On-time	Battery Power Backlight On-time	External Power Backlight On-time
0	No	**Yes**	No	No	No
5000	**Yes**	**Yes**	**Yes**	No	No
10000	**Yes**	**Yes**	**Yes**	**Yes**	No
15000	**Yes**	**Yes**	**Yes**	No	No
30000	**Yes**	**Yes**	**Yes**	**Yes**	No
60000	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
120000	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
180000	No	No	No	**Yes**	**Yes**
240000	No	No	No	**Yes**	**Yes**
300000	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
360000	No	No	No	No	**Yes**
420000	No	No	No	No	**Yes**
480000	No	No	No	No	**Yes**
540000	No	No	No	No	**Yes**
600000	No	No	No	No	**Yes**
2147483647 (Always On, maximum value)	**Yes**	No	**Yes**	No[1]	No[1]

[1] The setting for “Always On” in Windows Mobile is controlled by a checkbox that enables/disables the On-time parameter. If the checkbox is disabled, the On-time value is ignored and the backlight will remain on.

If a backlight timeout is set to one of the selectable values shown in the list, then the Control Panel applet will display the correct value for that timeout. On the other hand if a backlight timeout is set, using the Mobile Devices SDK, to a value that is not on the list of selectable values, then the Control Panel applet will display an empty box for that timeout setting.

On Psion Teklogix computers running Windows CE, the Display and Keyboard Backlight On-time values determine how long the display and keyboard backlights will remain on, respectively. Any user interaction (key press, touch screen press, scan, etc.) will reset both timeout counters back to the beginning. If there has been no user interaction by the end of the keyboard on-time duration, the keyboard backlight will turn off. At the end of the display on-time duration, the display backlight will dim to half intensity. If there is still no user interaction by the end of the display dim-time duration, then the display backlight will turn off completely.

On Psion Teklogix computers running Windows Mobile, the display and keyboard backlight time-outs are not controlled independently; instead both are controlled by a single timeout value. However, there are still two timeout values which may be set; one for when the computer is running on battery power, and one for when it is connected to a constant external power source.

5.2 Backlight Configuration Parameters

The following universal methods are available in all development languages for getting and setting backlight configuration values—see the API Reference Manuals for the name of the method in each development environment:

- Get a boolean setting.
- Get an integer setting
- Set a boolean setting.
- Set an integer setting

The following table lists the ranges of values and the default values available for each Psion Teklogix computer:

Parameter	Psion Teklogix Computer	Range Of Values	Default Value
Display threshold (%)	753x	0 to 100	71
	8515		N/A
	8525/8530		N/A
	Ikôn Windows CE		N/A
	Ikôn Windows Mobile		N/A
	NEO		N/A
	WORKABOUT PRO Windows CE		N/A
	WORKABOUT PRO Windows Mobile		N/A
Display intensity (%)	753x	0 to 100	80
	8515	0 to 100	48
	8525/8530	0 to 100	100
	Ikôn Windows CE	0 to 100	65
	Ikôn Windows Mobile	0 to 100	65
	NEO	0 to 100	47
	WORKABOUT PRO Windows CE	0 to 100	35
	WORKABOUT PRO Windows Mobile	0 to 100	35

Parameter	Psion Teklogix Computer	Range Of Values	Default Value
Display timeout (ms)	753x	0 to 2147483647	30000
	8515	0 to 2147483647	30000
	8525/8530	0 to 2147483647	30000
	Ikôn Windows CE	0 to 2147483647	30000
	Ikôn Windows Mobile	60000 to 600000	30000[1]
	NEO	0 to 2147483647	30000
	WORKABOUT PRO Windows CE	0 to 2147483647	30000
	WORKABOUT PRO Windows Mobile	60000 to 600000	30000[1]
Display dimtime (ms)	753x	0 to 2147483647	30000
	8515	0 to 2147483647	30000
	8525/8530	0 to 2147483647	30000
	Ikôn Windows CE	0 to 2147483647	30000
	Ikôn Windows Mobile		N/A
	NEO	0 to 2147483647	30000
	WORKABOUT PRO Windows CE	0 to 2147483647	30000
	WORKABOUT PRO Windows Mobile		N/A
Display always on for external power (boolean)	753x	True / False	True
	8515	True / False	True
	8525/8530	True / False	True
	Ikôn Windows CE	True / False	True
	Ikôn Windows Mobile	True / False	False
	NEO	True / False	True
	WORKABOUT PRO Windows CE	True / False	True
	WORKABOUT PRO Windows Mobile	True / False	False

Parameter	Psion Teklogix Computer	Range Of Values	Default Value
Keyboard threshold (%)	753x	0 to 100	0
	8515		N/A
	8525/8530		N/A
	Ikôn Windows CE		N/A
	Ikôn Windows Mobile		N/A
	NEO		N/A
	WORKABOUT PRO Windows CE		N/A
	WORKABOUT PRO Windows Mobile		N/A
Keyboard intensity (%)	753x	0 to 100	48
	8515	0 to 100	47
	8525/8530	0 to 100	48
	Ikôn Windows CE	0 to 100	50
	Ikôn Windows Mobile	0 to 100	50
	NEO	0 to 100	0
	WORKABOUT PRO Windows CE	0 to 100	40
	WORKABOUT PRO Windows Mobile	0 to 100	40
Keyboard timeout (ms)	753x	0 to 2147483647	15000
	8515	0 to 2147483647	15000
	8525/8530	0 to 2147483647	15000
	Ikôn Windows CE	0 to 2147483647	15000
	Ikôn Windows Mobile	60000 to 600000	30000[1]
	NEO	0 to 2147483647	15000
	WORKABOUT PRO Windows CE	0 to 2147483647	15000
	WORKABOUT PRO Windows Mobile	60000 to 600000	30000[1]

Parameter	Psion Teklogix Computer	Range Of Values	Default Value
Keyboard always on for external power (boolean)	753x	True / False	False
	8515	True / False	False
	8525/8530	True / False	False
	Ikôn Windows CE	True / False	False
	Ikôn Windows Mobile	True / False	False
	NEO	True / False	False
	WORKABOUT PRO Windows CE	True / False	False
	WORKABOUT PRO Windows Mobile	True / False	False

[1] On Windows Mobile devices, reading the default value returns the Battery Power Backlight On-time value of 30000. However, when setting the backlight value, it is written to the External Power Backlight On-time value which enforces the range of 60000 - 600000 for that parameter (1 minute to 10 minutes).

5.3 Backlight API Elements

C++: The backlights on all Psion Teklogix computers are controlled using the **PsionTeklogix::Backlight** namespace.

Java: The backlights on all Psion Teklogix computers are controlled using the **com.teklogix.backlight** package.

.NET: The backlights on all Psion Teklogix computers are controlled using the **PsionTeklogix.Backlight** namespace.

POWER MANAGEMENT 6

6.1 Battery Power Management 37
6.1.1 Suspend Timeout 37
6.1.2 Events. 38
6.1.3 Power-on Keyguard 38
6.1.4 Reset Key. 38
6.1.5 Battery Information. 38
6.1.5.1 Battery Threshold 38
6.1.5.2 Main Battery and Backup Battery Lifetimes And Remaining Charge 39
6.1.6 Power Management API Elements. 39
6.2 Card Slots. 39
6.2.1 Controlling Power Through The Windows Control Panel 41
6.2.2 Controlling Power Through The SDK 42
6.2.3 Card Slot Control API Elements 43
6.3 Peripherals 43
6.3.1 Power Modes 43
6.3.2 Power Modes API Elements 43

6.1 Battery Power Management

The Mobile Devices SDK provides features that can be used to control and preserve battery power in Psion Teklogix computers.

6.1.1 Suspend Timeout

The suspend timeout determines how long the Psion Teklogix computer will wait for any activity (key press, scan, or touch screen event, or a wireless radio transmission) before it should go into the suspend state. In the suspend state, the keyboard, display and all peripherals including the wireless radio are turned off, but the operating system remains active and any applications are maintained in their current state until the terminal resumes operation.

When the user wakes the computer — by pressing the **[ENTER]/Power** button — the device returns to the normal powered state. The backlight, the display, and all peripheral devices turn on and the suspend timer begins counting again.

The suspend timeout is configured as an integer variable measured in seconds. Any value between 0 and the maximum value of 2147483647 can be configured, however the Control Panel Power applet only allows a small number of pre-defined values to be set. If you set the suspend timeout using the SDK to a value not on the pre-defined list, the Control Panel Power applet will instead display the default value for that setting.

The list of pre-defined Control Panel Power applet suspend timeouts, and their corresponding integer values in seconds are available on Psion Teklogix computers depending on the operating system as follows:

		Windows CE 5.0		Windows Mobile	
Timeout	Int Value	Battery Power	External Power	Battery Power	External Power
Never	0	**Yes**	**Yes**	**Yes**	**Yes**
1 min	60	**Yes**	**Yes**	**Yes**	**Yes**
2 min	120	**Yes**	**Yes**	**Yes**	**Yes**
3 min	180	**Yes**	No	**Yes**	No
4 min	240	**Yes**	No	**Yes**	No
5 min	300	**Yes**	**Yes**	**Yes**	**Yes**
10 min	600	**Yes**	**Yes**	No	**Yes**
15 min	900	No	**Yes**	No	**Yes**
30 min	1800	**Yes**	**Yes**	No	**Yes**

6.1.2 Events

The following power state events are generated by the computer and can be detected by the Mobile Devices SDK:

- AC Power online
- AC Power offline
- Battery status change
- Power information change
- Power transition

6.1.3 Power-on Keyguard

The power-on keyguard feature specifies the number of key presses that are needed to power up the Psion Teklogix computer. Multiple key presses protect the hand-held computer from accidental power-up.

This option is only available on the WORKABOUT PRO with the Windows Mobile operating system. The Power-on Keyguard variable can be set to one of two values as follows:

- OneKey — User must press and hold the [ENTER] key for two seconds.
- TwoKey — User must press and release the [BLUE] key, then hold the [ENTER] key for two seconds.

6.1.4 Reset Key

On each Psion Teklogix computer there are default key combinations which can be used to reset power. The combinations can be enabled or disabled via the Mobile Devices SDK. The following options are available:

- BoostResetKey — Enable/disable boot to BooSt (bootstrap menu) reset key sequence. If disabled, the device will do a cold reset instead when the BooSt reset key sequence is performed.
- ColdResetKey — Enable/disable cold reset key sequence.
  On devices running Windows Mobile, this option is invalid; use WarmResetKey instead.
- WarmResetKey — Enable/disable warm reset key sequence. If disabled, the device will do nothing if the warm reset key sequence is performed.

6.1.5 Battery Information

6.1.5.1 Battery Threshold

Some Psion Teklogix computers have only a small backup battery, or no backup battery. On these computers, once the main battery has completely discharged, users have a very short time during which to change the battery before all volatile data is lost. In this situation, it can be desirable to have the device give the low battery warning prematurely, thus ensuring it gets replaced or recharged before data is lost.

Setting the battery suspend threshold forces the computer to suspend when that threshold is reached, before the main battery has fully discharged. The battery suspend threshold value is expressed as a percentage of battery charge remaining. The set method will accept any integer value from 0 to 20, however the actual range accepted by a given computer model may not actually go as high as 20; any value set above the terminal's maximum value will be stored as the maximum (e.g. setting a value of 18 on the **NEO** will result in a value of 13 being stored).

The Battery Suspend Threshold value ranges for each computer model are as follows:

Psion Teklogix Computer	Battery Suspend Threshold
753x	0 to 18
8515	Not available
8525/8530	Not available
Ikôn	0 to 8
NEO	0 to 13
WORKABOUT PRO	Not available
WORKABOUT PRO G2	0 to 17

6.1.5.2 Main Battery and Backup Battery Lifetimes And Remaining Charge

The battery lifetime reported by the Mobile Devices SDK may not be accurate as various system configurations, power management settings, and the activity of various peripheral devices all affect the rate that charge is drained from the battery.

It may not be possible to determine the remaining lifetime or the full lifetime of the backup battery while the computer is being powered by an external source, such as an AC adaptor. As a result, these calls may throw an exception if the computer is on external power.

The amount of charge reported as remaining in the battery may not be accurate. Also, this value can vary due to the activity of peripheral devices such as 802.11 or Narrowband radios.

6.1.6 Power Management API Elements

C++: The power on all Psion Teklogix computers is controlled using the **PsionTeklogix::PowerManagement** namespace.

Java: The power on all Psion Teklogix computers is controlled using the **com.teklogix.power** package.

.NET: The power on all Psion Teklogix computers is controlled using the **PsionTeklogix.Power** namespace.

6.2 Card Slots

The Mobile Devices SDK provides functions that control the power status of card slots. There are situations, such as in hospitals or airports, where it is necessary to temporarily prevent radios from transmitting. Using SDK functions to disable power to the card slot containing the radio achieves this. Most Psion Teklogix hand-held and vehicle-mount computers have this feature available on all card slots; however, there are some exceptions. Refer to the table below for details.

For those card slots that cannot be powered off, the driver for the card slot is unloaded, and all further attempts by the application to communicate with the card slot are ignored; however, the card slot remains fully powered, so the outcome depends on the behaviour of the radio card under these circumstances.

The following table lists the Psion Teklogix Windows CE-based computers, their card slots, and whether the card slots can be controlled by the SDK. The “SDK Hardware Name” is the specific string used with the get / set method to identify the card slot.

Psion Teklogix Computer	Card Slot Type	SDK Hardware Name	Software Control Of Power State	Cards Accepted
7530	SDIO	SD-MMC	Yes	SD card MMC card
	Compact Flash	PCMCIA Slot0	Yes	CF card
	PCMCIA	PCMCIA Slot1	Yes	Cardbus card PCMCIA card CF card in adaptor
7535	SD-MMC	SD-MMC	Yes[1]	SD card MMC card
	CF	PCMCIA Slot0	Yes[2]	CF card
[1] On the7535, the SD-MMC card slot cannot be powered off. The driver for the card will be unloaded and further activity on the slot will be ignored, but the device will be powered and may still be active in some way (e.g. a radio may still transmit/receive). [2] On the 7535, calls to the power state settijng method for PCMCIA Slot1 are ignored.				
8515	MicroSD	SD-MMC	No	MicroSD card (memory only)
	Compact Flash	PCMCIA Slot0	Yes	
8525/8530	SD-MMC	SD-MMC	Yes	SD card MMC card
	Compact Flash	PCMCIA Slot0	Yes	CF card
	PCMCIA	PCMCIA Slot1	Yes	PCMCIA card CF card in adaptor
Ikôn	MicroSD	SD-MMC	Yes	MicroSD card
	Proprietary	PCMCIA Slot0	Yes	WLAN Radio
NEO	MicroSD	SD-MMC	Yes	MicroSD card

Psion Teklogix Computer	Card Slot Type	SDK Hardware Name	Software Control Of Power State	Cards Accepted
WORKABOUT PRO	SD-MMC	SD-MMC	Yes	SD card (memory only) MMC card (memory only)
	CF	PCMCIA Slot0	Yes	CF card
	PCMCIA (upgradable on 100-pin connector)	PCMCIA Slot1[3]	Yes	PCMCIA card

[3] On the WORKABOUT PRO, when the PCMCIA slot is not installed, calls to the power state setting method for PCMCIA Slot1 throws an exception.

Power to software-controllable card slots is controlled using the following processes:

- Through the Windows CE Control Panel Power Properties window.
- Through application software using the SDK.

The power state can be set by either method. The most recently set state, from whichever source, applies.

6.2.1 Controlling Power Through The Windows Control Panel

Power to the software-controllable card slots can be controlled through the Windows Control Panel. Psion Teklogix hand-held computers have a Power icon on the Control Panel. Selecting this icon opens the *Power Properties* window, which has several different tabs. Select the *Card Slots* tab. For each card slot on the hand-held computer the following are listed:

- The name of the card slot.
- One of:
  - The name of the peripheral occupying the card slot.
  - Disabled.
  - Empty Slot.
- A checkbox indicating whether the card slot is enabled or disabled.

Click on the checkbox to toggle the power state of the card slot. Changes do not take effect until you click the **Apply** button.

Refer to the user manual for the Psion Teklogix hand-held or vehicle-mount computer for more details.

6.2.2 Controlling Power Through The SDK

Each of the API libraries has methods for the following (refer to the documentation for the relevant API library for more details)

Method Purpose	Input & Output
Set the power state	Input: <Hardware name> and <Power state>
Get the power state	Input: <Hardware name> Output: <Power state>

Where: <Hardware name> identifies the card slot. See "Card Slots" on page 39 for valid values.

<Power state> has one of the following values:
PowerState_Off
PowerState_On
PowerState_Unknown (only valid when querying the power state)

Querying The Power State

When the power state of a card slot is queried, the result is interpreted as follows:

- PowerState_Off — returned when there is no card in the card slot, or when there is a card in the slot and the card slot is powered down.
- PowerState_On — returned when there is a card in the slot and the card slot is powered on.
- PowerState_Unknown — returned when the power state of the slot cannot be determined.

Changing The Power State

When a slot is in PowerState_On, it is powered down by one of the following:

- Using the SDK method for setting the power state.
- Through the Windows Control Panel.

When a slot is in PowerState_Off, it is powered up by one of the following:

- Using the SDK method for setting the power state.
- Through the Windows Control Panel.
- Performing a clean start of the Windows operating system (see Section 1.5: "Resetting Psion Teklogix Computers").

Events

If there is a card in the card slot when the power state is changed, the following events are generated:

- Changing the state from PowerState_On to PowerState_Off generates a card removal event.
- Changing the state from PowerState_Off to PowerState_On generates a card insertion event.

While the card slot is in PowerState_On, the following events occur:

- Inserting a card generates a card insertion event.
- Removing a card generated a card removal event.

While the slot is in PowerState_Off, no card insertion or card removal events are generated.

6.2.3 Card Slot Control API Elements

C++: The card slots on all Psion Teklogix computers are controlled using the **PsionTeklogix::Peripherals** namespace.

Java: The card slots on all Psion Teklogix computers are controlled using the **Peripherals** class in the **com.teklogix.system** package.

.**NET:** The card slots on all Psion Teklogix computers are controlled using the **PsionTeklogix.Peripherals** namespace.

6.3 Peripherals

The Mobile Devices SDK provides functions that control the power status of peripherals that are attached to the Psion Teklogix computer.

6.3.1 Power Modes

The Mobile Devices SDK can detect and modify the manner in which power is applied to peripheral devices.

Querying the power state of a peripheral device returns one of the following constant values:

- AlwaysOff — is always powered down.
- AlwaysOn — is always powered up.
- OffAfterOnFor5Minutes — powers down after being powered up for 5 minutes.
- OffAfterOnFor15Minutes — powers down after being powered up for 15 minutes.
- OffAfterOnFor30Minutes — powers down after being powered up for 30 minutes.
- OffAfterOnFor60Minutes — powers down after being powered up for 60 minutes.
- OffAfterOnFor120Minutes — powers down after being powered up for 120 minutes.
- OffFollowSuspend — powers down after computer goes into suspend mode.

6.3.2 Power Modes API Elements

C++: The power modes on all Psion Teklogix computers is controlled using the **PsionTeklogix::Peripherals::PowerMode** namespace.

Java: The power on all Psion Teklogix computers is controlled using the **com.teklogix.system.PeripheralPowerMode** interface.

.NET: The power on all Psion Teklogix computers is controlled using the **PsionTeklogix.Peripherals.PowerMode** class.

7 DISPLAY

7.1 Display 47
7.2 Display API Elements 48

7.1 Display

The Mobile Devices SDK provides functions that obtain information on the display screen hardware that cannot be easily obtained using standard features of the development languages. Display features are available as follows:

Psion Teklogix Computer	Screen Size	Number of Colours	Touchscreen	Type
7530	• 240 pixels wide • 320 pixels high • ¼ VGA • 3.5 in. diagonal	65536	Optional	Reflective
7535	• 240 pixels wide • 320 pixels high • ¼ VGA • 3.5 in. diagonal	65536	Optional	Reflective
8515	• 640 pixels wide • 480 pixels high • VGA • 6.4 in. diagonal	65536	Yes	Reflective
8525	• 640 pixels wide • 240 pixels high • ½ VGA • 8.8 in. diagonal	65536	Yes	Transmissive
8530	• 800 pixels wide • 600 pixels high • SVGA • 10.4 in diagonal	65536	Yes	Reflective
Ikôn	• 480 pixels wide • 640 pixels high • VGA • 3.7 in. diagonal	65536	Yes	Reflective
NEO	• 240 pixels wide • 320 pixels high • ¼ VGA • 2.7 in. diagonal	65536	Yes	Transmissive
WORKABOUT PRO	• 480 pixels wide • 640 pixels high • VGA • 3.6 in. diagonal	65536	Yes	Reflective

Refer to the user manuals for the respective Psion Teklogix computers for more information.

The following display information can be retrieved using the SDK:

Display Hardware Property	C++	Java	.NET
Colour or monochrome	No	Yes	Yes
Display type	No	Yes	Yes
Display dimensions in pixels	No	Yes	Yes
Display dimensions in millimetres	No	Yes	Yes
Maximum number of colours, or shades of grey	No	Yes	Yes
Touchscreen, or non-touchscreen	Yes	Yes	Yes

The following properties are returned, as name/value pairs, by the Mobile Devices SDK:

Property Name String	Value Type	Value
Display Type	String	Transmissive, Reflective, Transreflective, or Unknown
Physical Width	Integer	Display width in millimetres
Physical Height	Integer	Display height in millimetres
Colour Display	Boolean	True, or False
Touch Screen Installed	Boolean	True, or False
Width in Pixels	Integer	Display width in pixels
Height in Pixels	Integer	Display height in pixels
Maximum Colours	Integer	Number of colours, or shades of grey available

7.2 Display API Elements

C++: Information concerning the display on all Psion Teklogix Windows CE computers is retrieved using the **PsionTeklogix::DisplayInformation** namespace. Additional display details are obtained using the Windows CE User Interface Services GDI function GetDeviceCaps ().

Java Information on the display hardware on all Psion Teklogix Windows CE computers is retrieved using the **com.teklogix.display** package.

.NET: Information on the display hardware on all Psion Teklogix Windows CE computers is retrieved using the **System.Windows.Forms** namespace in the .NET Compact Framework, or using the **PsionTeklogix.SystemPTX.DisplayInformation** class.

8 INDICATORS

8.1 Indicators 51
8.2 Using LED Colours 51
8.3 Controlling Pulses 52
8.4 Controlling Illumination Patterns 52
8.5 Indicator API Elements. 52

8.1 Indicators

Most Psion Teklogix computers have LEDs that can be controlled by applications. Typically, these are used to indicate device activity, data reception, data transmission, error conditions, alerts, and software updates. LEDs are available as follows:

Psion Teklogix Computer	Number Of LEDs	Type Of LEDs	Number Of LEDs Available For Applications	LED Names
7530	4	Tricolour	1	Application
7535	4	Tricolour	1	Application
8515	1	Single colour	0	N/A
8525/8530	4	Tricolour	1	Application
Ikôn	3	Single colour	1	Application
NEO	1	Tri colour	1	Application
WORKABOUT PRO	1	Tricolour	1	Application

8.2 Using LED Colours

Each LED emits one or more colours. Some colours are built into the LED. Other colours are created by illuminating two or more built-in colours at the same time. The following terms are used to distinguish these colours:

Component colour: This colour is built into the LED.

Composite colour: This colour is created by illuminating two or more component colours at the same time on the LED.

Default on colour: A LED is illuminated with the default on colour, when it is illuminated without explicitly naming an illumination colour.

Available colour: This can be either a component colour or a composite colour.

LED colours are available on Psion Teklogix computers as follows:

Psion Teklogix Computer	LED Name	Component Colors	Composite Colours	Default On Colour
7530	Application	Red Green	Yellow	Green
7535	Application	Red Green	Yellow	Green
8525/8530	Application	Red Green	Yellow	Green
Ikôn	Application	Yellow	Yellow	Yellow
NEO	Application	Red Green	Yellow	Green
WORKABOUT PRO	Application	Red Green	Yellow	Green

8.3 Controlling Pulses

The following terms are used to describe the behaviour of a pulsing LED:

Independent colour: On multi-colour LEDs, if the independent colour flag is set, this pulse can add to the colour being displayed by the LED.

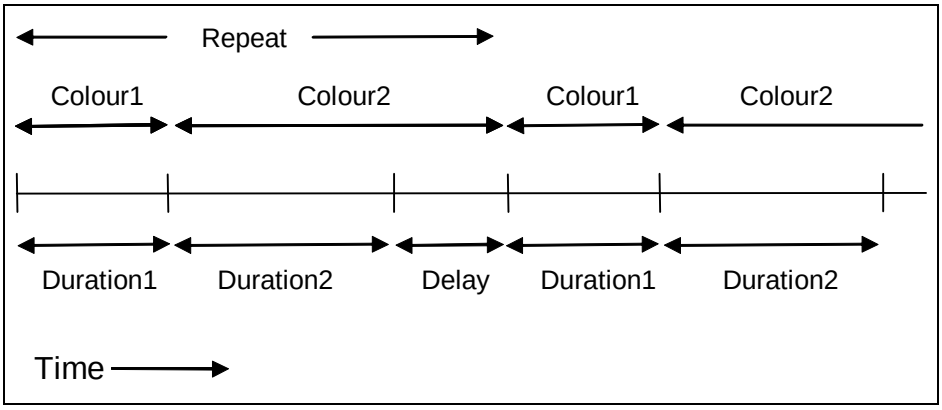
Extend current pulse: This feature only applies when the independent colour flag is also set. If the LED is in the process of performing a pulse using the same colour, the duration of this new pulse replaces the duration of the existing pulse. This can result in the duration either being extended or reduced.

Several overlapping pulses can exist. The effects depend on the settings of the independent colour flag and the extend current pulse flag.

8.4 Controlling Illumination Patterns

The Mobile Devices SDK provides the ability to display a two-colour illumination pattern on a LED.

The following diagram shows how an illumination pattern is structured:



The two colours, their duration times, the delay between repeats, and the number of repeats can be defined in the application. See the API Reference manual for your programming language for information on how to specify these values.

8.5 Indicator API Elements

C++: The display on all Psion Teklogix computers is controlled using the **PsionTeklogix::Indicators** namespace.

Java: The display on all Psion Teklogix computers is controlled using the **com.teklogix.indicators** package.

.NET: The display on all Psion Teklogix computers is controlled using the **PsionTeklogix.Indicators** namespace.

KEYBOARD 9

9.1 Keyboard 55
9.1.1 Supported Keyboards. 55
9.1.1 Supported Keyboards. 55
9.1.2 Disabling The Keyboard 56
9.1.3 Keyboard API Elements 56
9.2 Keyboard Remapping 56
9.2.1 Scan Codes For Psion Teklogix Keyboards 58
9.2.2 Keyboard Remapping API Elements. 62
9.3 Key Insertion 62
9.3.1 Key Insertion API Elements 63

9.1 Keyboard

Microsoft Windows maintains a device-independent keyboard model that enables it to support a variety of keyboards. At the lowest level, each key on the keyboard generates a scan code when the key is pressed and released. The scan code is a hardware-dependent number that identifies the physical location of the key on the keyboard. Unlike Windows-based desktop operating systems, Windows CE has no standard set of keyboard scan codes. The keyboard driver translates or maps each scan code to a virtual key code. The virtual key code is a hardware-independent number that identifies the key to be sent to the application.

Because keyboard layouts vary between spoken languages, Windows offers only the core set of virtual key codes that are found on all keyboards. This core set includes English characters, numbers, and a few critical keys, such as the function and arrow keys.

In addition to mapping, the Windows keyboard driver determines which characters the virtual key generates. A single virtual key can generate different characters depending on the state of modifier keys, such as the [SHIFT] and [CTRL] keys.

The Mobile Devices SDK provides support for the Psion Teklogix-specific keys.

9.1.1 Supported Keyboards

When the keyboard type is queried through the Mobile Devices SDK, the following strings may be returned:

Psion Teklogix Computer	Keyboard Description Strings
7530 / 7535	36-Key
	37-Key
	58-Key
	63-Key
	None
8515 / 8525 / 8530	68-Key ABC
	68-Key Azerty
	68-Key Qwerty
	Unknown
	None
Ikôn	28-Key WinCE
	28-Key WM
	28-Key WM Phone
	47-Key
	Unknown

Psion Teklogix Computer	Keyboard Description Strings
NEO	26-Key
	48-Key
	Unknown
WORKABOUT PRO G1	24-Key Keyboard
	52-Key Keyboard
WORKABOUT PRO G2	25-key
	52-key
	Unknown

9.1.2 Disabling The Keyboard

The keyboard can be disabled at the hardware level. When disabled, no key presses are recorded. Disabling the keyboard may be used to prevent user data entry while a program is performing a critical operation such as a database transaction or a screen refresh, when a key press could cause problems.

Important: ***Take great care when disabling the keyboard. If an application terminates while the keyboard is disabled, there is no easy way to re-enable the keyboard. This can leave the computer in an unusable state (the touchscreen is still active, assuming the computer has one).***

Some keyboard operations are still available even when the keyboard is disabled, including resets, and placing the computer into suspend mode.

9.1.3 Keyboard API Elements

C++: The display on all Psion Teklogix computers is controlled using the **PsionTeklogix::Keyboard** namespace.

Java: The display on all Psion Teklogix computers is controlled using the **com.teklogix.keyboard** package.

.NET: The display on all Psion Teklogix computers is controlled using the **PsionTeklogix.Keyboard** namespace.

9.2 Keyboard Remapping

The key stroke information sent to an application when a key is pressed can be altered through a process of remapping key code values. There are two sets of key codes — scan codes and virtual key codes — which define the associations between a physical key pressed, and the key value that is sent to an application. The default associations of these key code sets characterize the normal behaviour of a particular keyboard. Keyboard remapping overrides the default behaviour of the keyboard keys.

A **scan code** is an integer value representing a key on a keyboard. Psion Teklogix computers use a hardware-independent list of scan codes that applies to all device models. See "Scan Codes For Psion Teklogix Keyboards" on page 58 for a list of these scan codes.

All Psion Teklogix computers have non-chorded keyboards. A non-chorded keyboard is a keyboard that does not handle simultaneous key presses. Each key pressed generates a unique scan code which is not modified by the state of other keys on the keyboard.

Modifier keys are keys on the keyboard that when pressed and released set a mode that can change the behaviour of other keys on the keyboard. The following keys are modifier keys: [BLUE],[ORANGE], [ALT], [CTRL], and [SHIFT]. These can change the virtual key code value generated by a subsequent scan code.

A **virtual key code** is a device-independent value defined by the system that identifies the purpose of a key, and is what informs an application about which key was pressed. Scan codes are mapped to virtual key codes by the keyboard driver. A single scan code can map to multiple virtual key codes, dependent on the current state of the [ORANGE], [BLUE] and [SHIFT] modifier keys.

The mapping between the scan codes and the virtual key codes is defined in a set of tables. There are separate tables to define the code mappings for normal operation, and for when the [SHIFT], [ORANGE] or [BLUE] modifiers are active. However, only the [ORANGE] and [BLUE] tables can be remapped, the [SHIFT] table cannot.

The [CTRL] and [ALT] modifier keys do not change the virtual key code generated, thus there are no key code tables for those modifiers. However, an application can detect the state of these modifiers upon receiving a virtual key code, and change its behaviour accordingly.

A **function** in keyboard remapping terminology is an operation that is performed when a particular scan code is generated by a key press. This operation may modify the virtual key code generated, or cause some other effect such as changing the backlight intensity. The following types of functions are available:

- Macro—maps a scan code to a macro key, which is then mapped into a sequence of one of more virtual key codes. No virtual key code is generated (other than those defined in the macro sequence).
- Operation only—maps a scan code to some specific behaviour (e.g. backlight intensity). No virtual key code is generated.
- Modifier key mapping—causes a scan code to simulate the pressing of a modifier key, in order to correctly update the modifier key state. Normal sequence for modifier keys is: off -> one shot -> locked -> off).
- Virtual key (+modifier)—maps a scan code to a virtual key code, and may simulate the pressing of one or more modifier keys.
- Direct Unicode mapping—maps a scan code directly to a Unicode character. This enables characters to be generated which have no virtual key equivalents., such as accented characters.
- Null mapping—causes a scan code to be ignored.

A scan code mapping can involve all of these elements. A scan code generated by a key can be mapped to a function, and possibly also to a virtual key, macro index, or Unicode character value.

Some characters do not have virtual key codes, but can be generated using shifted-key codes. For example, a '+' character is actually generated by sending a shifted '=' virtual key code (that is, the scan code is mapped to **VK_EQUAL** and the function **Function.SendShiftedCode**). These mappings can be inferred from a standard PC keyboard.

If two threads or processes attempt to modify the keyboard scan code mappings at the same time, the results are undefined.

Scan code remapping enables applications to perform the following operations:

- Create one or more scan code remappings for a scan code table.
- Remove a scan code remapping from a scan code table.
- Remove all scan code remappings from a scan code table.
- Check to see if a particular scan code has been remapped.
- Convert the table to a printable string.

There are three tables where scan codes can be remapped:

- Normal—remappings for all scan codes when neither the [BLUE] nor the [ORANGE] keys are pressed.
- Blue—remappings for when the [BLUE] key is pressed.
- Orange—remappings for when the [ORANGE] key is pressed.

If both the [BLUE] and [ORANGE] keys have been pressed (they are both in either the one-shot or locked state), the remapping for the [BLUE] key has precedence.

9.2.1 Scan Codes For Psion Teklogix Keyboards

The following table lists the hardware-independent scan codes used on all Psion Teklogix keyboards. Not all keys / scan codes are available on all computers — for instance, the scan codes for phone keys only apply to devices with cellular phone capability.

Key	Scan Code Name	Scan Code
0	hwIndepScanCode_0	0
1	hwIndepScanCode_1	1
2	hwIndepScanCode_2	2
3	hwIndepScanCode_3	3
4	hwIndepScanCode_4	4
5	hwIndepScanCode_5	5
6	hwIndepScanCode_6	6
7	hwIndepScanCode_7	7
8	hwIndepScanCode_8	8
9	hwIndepScanCode_9	9
A	hwIndepScanCode_A	10
B	hwIndepScanCode_B	11
C	hwIndepScanCode_C	12
D	hwIndepScanCode_D	13
E	hwIndepScanCode_E	14
F	hwIndepScanCode_F	15
G	hwIndepScanCode_G	16

Key	Scan Code Name	Scan Code
H	hwIndepScanCode_H	17
I	hwIndepScanCode_I	18
J	hwIndepScanCode_J	19
K	hwIndepScanCode_K	20
L	hwIndepScanCode_L	21
M	hwIndepScanCode_M	22
N	hwIndepScanCode_N	23
O	hwIndepScanCode_O	24
P	hwIndepScanCode_P	25
Q	hwIndepScanCode_Q	26
R	hwIndepScanCode_R	27
S	hwIndepScanCode_S	28
T	hwIndepScanCode_T	29
U	hwIndepScanCode_U	30
V	hwIndepScanCode_V	31
W	hwIndepScanCode_W	32
X	hwIndepScanCode_X	33
Y	hwIndepScanCode_Y	34
Z	hwIndepScanCode_Z	35
Function 1	hwIndepScanCode_F1	40
Function 2	hwIndepScanCode_F2	41
Function 3	hwIndepScanCode_F3	42
Function 4	hwIndepScanCode_F4	43
Function 5	hwIndepScanCode_F5	44
Function 6	hwIndepScanCode_F6	45
Function 7	hwIndepScanCode_F7	46
Function 8	hwIndepScanCode_F8	47
Function 9	hwIndepScanCode_F9	48
Function 10	hwIndepScanCode_F10	49
Function 11	hwIndepScanCode_F11	50

Key	Scan Code Name	Scan Code
Function 12	hwIndepScanCode_F12	51
Function 13	hwIndepScanCode_F13	52
Function 14	hwIndepScanCode_F14	53
Function 15	hwIndepScanCode_F15	54
Function 16	hwIndepScanCode_F16	55
Function 17	hwIndepScanCode_F17	56
Function 18	hwIndepScanCode_F18	57
Function 19	hwIndepScanCode_F19	58
Function 20	hwIndepScanCode_F20	59
Function 21	hwIndepScanCode_F21	60
Function 22	hwIndepScanCode_F22	61
Function 23	hwIndepScanCode_F23	62
Function 24	hwIndepScanCode_F24	63
Function 25	hwIndepScanCode_F25	64
Function 26	hwIndepScanCode_F26	65
Function 27	hwIndepScanCode_F27	66
Function 28	hwIndepScanCode_F28	67
Function 29	hwIndepScanCode_F29	68
Function 30	hwIndepScanCode_F30	69
Macro 1	hwIndepScanCode_M1	70
Macro 2	hwIndepScanCode_M2	71
Macro 3	hwIndepScanCode_M3	72
Macro 4	hwIndepScanCode_M4	73
Macro 5	hwIndepScanCode_M5	74
Macro 6	hwIndepScanCode_M6	75
Macro 7	hwIndepScanCode_M7	76
Macro 8	hwIndepScanCode_M8	77
Macro 9	hwIndepScanCode_M9	78
Macro 10	hwIndepScanCode_M10	79
Macro 11	hwIndepScanCode_M11	80

Key	Scan Code Name	Scan Code
Macro 12	hwIndepScanCode_M12	81
Scan	hwIndepScanCode_Scan	90
Pistol Grip Trigger	hwIndepScanCode_HandgripScan	91
Left Scan	hwIndepScanCode_LeftScan	92
Right Scan	hwIndepScanCode_RightScan	93
Left Scan, Side	hwIndepScanCode_SideLeftScan	94
Right Scan, Side	hwIndepScanCode_SideRightScan	95
Alt	hwIndepScanCode_Alt	100
Backspace	hwIndepScanCode_Bksp	101
Blue	hwIndepScanCode_Blue	102
Backlight	hwIndepScanCode_Backlight	103
Comma	hwIndepScanCode_Comma	104
ContrastUp	hwIndepScanCode_ContrastUp	105
Ctrl	hwIndepScanCode_Ctrl	106
Delete	hwIndepScanCode_Del	107
Enter	hwIndepScanCode_Enter	108
End	hwIndepScanCode_End	109
Esc	hwIndepScanCode_Esc	110
Home	hwIndepScanCode_Home	111
Insert	hwIndepScanCode_Ins	112
Orange	hwIndepScanCode_Orange	113
Period	hwIndepScanCode_Period	114
PgDn	hwIndepScanCode_PgDn	115
PgUp	hwIndepScanCode_PgUp	116
Shift	hwIndepScanCode_Shift	117
Space	hwIndepScanCode_Space	118
Tab	hwIndepScanCode_Tab	119
Down Arrow	hwIndepScanCode_DownArrow	120
Left Arrow	hwIndepScanCode_LeftArrow	121
Right Arrow	hwIndepScanCode_RightArrow	122

Key	Scan Code Name	Scan Code
Up Arrow	hwIndepScanCode_UpArrow	123
Left Down Arrow, Side	hwIndepScanCode_SideLeftDownArrow	124
Left Up Arrow, Side	hwIndepScanCode_SideLeftUpArrow	125
Right Down Arrow, Side	hwIndepScanCode_SideRightDownArrow	126
Right Up Arrow, Side	hwIndepScanCode_SideRightUpArrow	127
Windows	hwIndepScanCode_Windows	128
Soft Key 1	hwIndepScanCode_SoftKey1	129
Soft Key 2	hwIndepScanCode_SoftKey2	130
Phone Send	hwIndepScanCode_PhoneSend	131
Phone End	hwIndepScanCode_PhoneEnd	132
Phone * (Star)	hwIndepScanCode_PhoneStar	133
Phone # (Pound)	hwIndepScanCode_PhonePound	134
Power	hwIndepScanCode_Power	135

9.2.2 Keyboard Remapping API Elements

C++: The keyboard remapping on all Psion Teklogix computers is controlled using the **PsionTeklogix::Keyboard::NoChording** namespace.

Java: The keyboard remapping on all Psion Teklogix computers is controlled using the **com.teklogix.keyboard.nochording** package.

.NET: The keyboard remapping on all Psion Teklogix computers is controlled using the **PsionTeklogix.Keyboard.NoChording** namespace.

9.3 Key Insertion

Key insertion permits a command key or a modifier key, with another optional related key, to be inserted into an input field. The following command keys and modifier keys can be inserted:

- Blue
- Orange
- Shift
- Control
- Alt
- Send unshifted code
- Send shifted code
- Contrast up
- Contrast down
- Volume up
- Volume down
- Scanner on
- Terminal off
- Backlight cycle up
- Macro
- Send unicode
- Backlight brighter
- Backlight dimmer
- Backlight cycle down
- System power state
- Send DPad code
- Trigger

Before this feature is invoked, the focus must be on the relevant input field.

This feature is typically used for the following, described in further detail below:

- In application lock-down mode, displaying key presses in alpha mode.
- Reversing an accidental press of the [BLUE] key or the [ORANGE] key.
- As a keyboard wedge.

Application Lock-Down Mode

When an application is operating in lock-down mode, the Windows task bar is not visible. Normally, when a user is entering alpha characters on a computer with a numeric keyboard ([2ABC], [3DEF], etc.), the intermediate characters are displayed on the Windows task bar until the desired character is selected. Key insertion allows an application to display the intermediate alphabetic characters directly in the input field.

Reversing Accidental Key Presses

If the [BLUE] or [ORANGE] key is accidently pressed by an operator during data entry, the results can be unexpected and can cause an input error. Detecting the accidental modifier key press, and reversing it within the application, ensures that the intended data is entered.

The Mobile Devices SDK provides functions that allow the [BLUE] key and the [ORANGE] key presses to be intercepted. The key insertion feature allows the application to reverse the setting of the key.

Keyboard Wedge

A keyboard wedge inserts characters into a field that is in focus. A single virtual key can be inserted into an input field by each call to the key insertion function. A command key, such as **Send unshifted code** accompanied by a virtual key code, wedges the associated virtual key into the input field.

9.3.1 Key Insertion API Elements

C++: Key insertion on all Psion Teklogix Windows computers is controlled using the **PsionTeklogix::Keyboard** namespace.

Java: Key insertion on all Psion Teklogix Windows computers is controlled using the **Keyboard** class in the **com.teklogix.keyboard** package.

.NET: Key insertion on all Psion Teklogix Windows computers is controlled using the **Keyboard** class in the **PsionTeklogix.Keyboard** namespace.

PERIPHERALS 10

10.1 Overview 67
10.2 Definition Of Terms 67
10.3 Events 67
10.4 Peripheral and Card Slot Power 67
10.5 Docking Station 67
10.6 Tether Ports 68
10.7 Peripheral API Elements 68

10.1 Overview

The Mobile Devices SDK enables applications to detect and control peripherals attached to Psion Teklogix computers, such as docking stations, tethered devices, and cards inserted in card slots.

10.2 Definition Of Terms

Some terms used in the chapter have precise definitions. They are defined in this section.

Adaptor: This is a hardware component that supports the connection of the computer to a network or a peripheral device. An adaptor can be a printed circuit board, a PC card, or circuitry that is part of the mother board.

Device driver: This is a software component that permits a computer system to communicate with a device. In most cases, the driver also manipulates the hardware in order to transmit the data to the device.

Peripheral or peripheral device: A device, such as a hard drive, printer, radio or modem, that is connected to a computer and is controlled by the computer's microprocessor.

10.3 Events

The following peripheral event types are detected by the Mobile Devices SDK:

- Adaptor event
- Docking station event
- Interface event
- Tether port event

Adaptor event: Occurs when the adaptor is connected to or removed from the slot.

Docking station event: Occurs when the device is inserted into or removed from the docking station.

Interface event: Occurs when the device is connected to or removed from the slot/port.

Tether port event: Occurs when the device is connected to or removed from the tether port.

10.4 Peripheral and Card Slot Power

For information on querying and controlling the power status of peripherals see Section 6.3: "Peripherals".

For information on controlling the power status of card slots, see Section 6.2: "Card Slots".

10.5 Docking Station

The Mobile Devices SDK can detect the type of docking station the Psion Teklogix computer is currently resting in. A docking station is an external hardware component. It can be one of the following:

- Portable docking module
- Battery charger
- Cradle

A docking station can include one or more additional serial ports, and USB ports.

10.6 Tether Ports

The Mobile Devices SDK can detect the type of peripheral device that is attached to the computer via an external tether port. It can also detect the attachment and removal of a tether port device.

Tether ports are available as follows:

Psion Teklogix Computer	Has A Tether Port
7530	Yes
7535	Optional
8515	Yes
8525/8530	Yes
Ikôn	Yes
NEO	Yes
WORKABOUT PRO	Optional

The following types of device can be attached to a tether port:

- Scanners
- RFID readers
- Imagers

10.7 Peripheral API Elements

C++: The peripherals on all Psion Teklogix computers is controlled using the **PsionTeklogix::System::Peripherals** namespace.

Java: The peripherals on all Psion Teklogix computers is controlled using the **com.teklogix.system** package.

.NET: The peripherals on all Psion Teklogix computers is controlled using the **PsionTeklogix.Peripherals** namespace.

PERMANENT STORAGE 11

11.1 Permanent Storage 71
11.2 Locking Permanent Storage 71
11.3 Permanent Storage API Elements 71

11.1 Permanent Storage

A small amount of permanent storage is provided on some Psion Teklogix computers. This storage is accessible through the Mobile Devices SDK, but is not accessible through the Control Panel. Data stored in permanent storage persists across power failure, all types of reset, and through clean starts.

There is only one permanent memory location in each device; multiple applications on the same device will not be able to use this memory for application-specific data.Typically, the permanent storage is used for assigning customized serial numbers to each device or for asset tracking purposes.

Important: ***The hardware used for permanent storage typically has a limited cycle life. Repeated write operations may cause the memory to become corrupt and unreliable. It is therefore recommended that this memory storage not be used for data that is expected to change frequently.***

The following table shows what Psion Teklogix computers have permanent memory storage, and how much is available.

Computer	Permanent Memory Available
WORKABOUT PRO	28 bytes
NEO	32 bytes

11.2 Locking Permanent Storage

The Mobile Devices SDK provides an application with the ability to lock the permanent storage area. **There is no unlock ability.**

Warning: ***To unlock the permanent storage, the computer must be returned to the Psion Teklogix repair depot. This cannot be a warranty repair.***

11.3 Permanent Storage API Elements

C++: Permanent storage on all Psion Teklogix Windows computers is controlled using the **PsionTeklogix::System::SystemInformation** namespace.

Java: Permanent storage on all Psion Teklogix Windows computers is controlled using the **SystemInformation** class in the **com.teklogix.system** package.

.NET: Permanent storage on all Psion Teklogix Windows computers is controlled using the **SystemInformation** class in the **PsionTeklogix.SystemPTX** namespace.

RAS (REMOTE ACCESS SERVICE) 12

12.1 Overview 75
12.2 RAS Architecture 75
12.3 RAS on Windows Mobile (Connection Manager) 76
12.4 RAS API Elements 76

12.1 Overview

Microsoft Windows CE .NET and Windows Mobile provide support for a Remote Access Service (RAS) server. You can use RAS in any computing environment that has a wide area network (WAN) or a virtual private network (VPN). RAS makes it possible to connect a remote client computer to a network server over a WAN or a VPN.

A Windows-based device can function as a RAS server that allows clients to connect to it using a WAN connection.

A Windows-based RAS server can accept connections through any Network Driver Interface Specification (NDIS) miniport in a WAN, including the Point-to-Point-Tunneling Protocol (PPTP) and AsyncMAC miniports. AsyncMAC is an NDIS WAN miniport driver that manages the sending and receiving of packets over TAPI devices. These devices include direct serial and modem connections.

The RAS server implementation supports authentication through the Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP) and Microsoft Challenge Handshake Authentication Protocol (MSCHAP). The Point-to-Point Protocol (PPP) supports 128-bit and 40-bit encryption.

In Windows, a RAS server is configured through the registry and I/O control codes (IOCTLs). Registry settings are used for boot-time configuration, and the IOCTLs are used for dynamic information.

Unlike Windows 2000, the Windows versions supported by Psion Teklogix computers do not support server callback or multilink connections. These Windows versions also do not support user domains. In user credentials, you can specify the user name and password, but not the domain name.

12.2 RAS Architecture

In the Windows networking architecture, the Windows-based device functioning as a RAS server communicates directly with PPP. This corresponds with the underlying WAN miniport — either PPTP or AsyncMAC — through Transmission Control Protocol/Internet Protocol (TCP/IP). When PPP receives requests to send IP packets from TCP/IP, it passes the packet on to the AsyncMAC miniport. After receiving a packet from PPP, the AsyncMAC miniport performs the asynchronous framing, and then forwards the packet to the TAPI device by calling Microsoft Win32 serial APIs. When receiving a packet through the network, the AsyncMAC miniport strips the asynchronous framing off the packet, verifies the Cyclic Redundancy Check (CRC), and passes the packet to PPP through the NDIS layer.

In a VPN, the PPTP WAN miniport communicates directly with TCP/IP. After receiving a packet that is addressed to the private network from TCP/IP, PPP performs the framing, and then forwards the packet to the PPTP WAN miniport. The PPTP WAN miniport encapsulates the information contained in the packet header, and then reroutes the packet back to TCP/IP. IP then attaches another header that contains the address of the PPTP server to the packet, and passes the packet on to PPP. PPP forwards the packet to either the AsyncMAC WAN miniport or a local area network (LAN) adaptor, such as an NE2000 adaptor. After receiving the packet over the network, the PPTP server strips the PPTP header off the packet, and then passes the packet to PPP.

12.3 RAS on Windows Mobile (Connection Manager)

In addition to RAS, Windows Mobile devices come with a Connection Manager which some applications, such as Internet Explorer, use to determine if there is an Internet connection. If an Internet connection is made using the Psion Teklogix RAS API, the Connection Manager will not be aware of it, thus any applications that rely on the Connection Manager will not be able to connect to the Internet through it.

Microsoft provides an API for creating connections through the Connection Manager. For more information, visit the following URLs:

http://msdn2.microsoft.com/en-us/library/bb416435.aspx
http://msdn2.microsoft.com/en-us/library/bb840031.aspx

Applications using direct socket connections will still be able to connect using a RAS connection on both Windows CE and Windows Mobile devices.

12.4 RAS API Elements

C++: The RAS server on all Psion Teklogix Windows computers is controlled using the Microsoft RAS Win32 APIs.

Java: The RAS server on all Psion Teklogix Windows computers is controlled using the **com.teklogix.ras** package.

.NET: The RAS server on all Psion Teklogix Windows computers is controlled using the **PsionTeklogix.RAS** namespace.

REMOTE SOCKETS 13

13.1 Overview 79
13.2 802.11 Systems 79
13.3 Narrowband Radio Systems 79
13.3.1 Remote Sockets Proxy 79
13.3.2 Starting A Remote Sockets Connection 79
13.3.3 Ending A Remote Sockets Connection 79
13.3.4 Configuring A Psion Teklogix Network Controller (9400/9500) 80
13.3.5 Configuring A Psion Teklogix Computer. 80
13.3.6 Troubleshooting A Remote Sockets Connection 80
13.3.7 Remote Sockets API Elements 82
13.4 Technical References 82

13.1 Overview

Sockets permit applications to communicate across a network with no dependence on the network protocol. Psion Teklogix Remote Sockets is available for narrowband systems on the following devices:

- 7530
- 8525
- 8530

The Remote Sockets API is only available for programming in C++, not in Java or .NET.

13.2 802.11 Systems

On Psion Teklogix 802.11 systems running Windows CE, the Windows Sockets 2 API is available from Microsoft. For further information refer to the following links on MSDN:
http://msdn.microsoft.com/en-us/library/aa454342.aspx
http://msdn.microsoft.com/en-us/library/ms740673.aspx.

13.3 Narrowband Radio Systems

On a Psion Teklogix narrowband radio system, communication between a Psion Teklogix computer and the network controller uses a proprietary radio protocol. Remote sockets enable a Windows CE-based Psion Teklogix computer to communicate with a remote host using sockets. This implementation is a reduced Berkeley Software Distribution-style (BSD) interface.

Remote sockets require a controller-based Psion Teklogix system implementation. Remote sockets are implemented using a proxy mechanism. Any socket calls made by the Psion Teklogix computer are directed to the network controller. The network controller in turn makes a native socket call to the remote host. The results of the proxy call are returned to the Psion Teklogix computer by the network controller. This mechanism allows the Psion Teklogix computer to appear to be using Windows sockets over a narrowband radio system.

This implementation has several advantages. Remote sockets allow Psion Teklogix computers to connect to hosts, and perform many functions that would not normally be available on a narrowband system. The remote sockets API provides a consistent, well-known interface familiar to software developers. The remote sockets protocol reduces the bandwidth required to less than is needed on a wired sockets implementation. The use of remote sockets also places much of the processing duties on the controller, which can handle multiple sockets and high volumes of data better than the hand-held or vehicle-mounted computer.

13.3.1 Remote Sockets Proxy

The network controller runs the Remote Sockets Proxy software. This interfaces with the remote host on behalf of Psion Teklogix hand-held computers, and relays information between them.

13.3.2 Starting A Remote Sockets Connection

PsionTeklogix::RemoteSockets::startup initializes the remote sockets feature and must be called before any other API element.

13.3.3 Ending A Remote Sockets Connection

To end a remote sockets connection, call the **PsionTeklogix::RemoteSockets::terminate** function. This function must be called before the application terminates.

13.3.4 Configuring A Psion Teklogix Network Controller (9400/9500)

On the Psion Teklogix network controller, each Psion Teklogix computer is considered to be a terminal. On the Psion Teklogix controller, navigate to the Radio Link Features screen, and set the following parameters:

- Select **Operate in Cellular Mode.**
- Set the automatic radio address assignment range.

  This range must be wide enough to accommodate all the sockets that can be opened on all the Psion Teklogix computers connecting through the controller. Each Psion Teklogix computer can open several sockets. Make this range large enough to allow for additional Psion Teklogix computers to be added later. Calculate the radio address range as follows:

  The **minimum** address range = (number of terminals) x (maximum number of sockets open on one terminal)

- Set the terminal number range.

  This range should be the same width as the automatic radio address assignment range.
- Save the configuration and reboot the controller.

13.3.5 Configuring A Psion Teklogix Computer

Psion Teklogix computers do not require any special configuration in order to use remote sockets.

13.3.6 Troubleshooting A Remote Sockets Connection

In most respects, the Psion Teklogix remote sockets implementation behaves like a Windows Sockets 2 implementation; however, since this is a two-stage process with the network controller acting as an intermediary, the following issues must be considered:

- Psion Teklogix computers may go offline / out of range / inactive for significant periods of time.

  The remote host should not close the sockets connection during these periods of time. If it does, the Psion Teklogix computer must close any open sockets and reopen them. Recovery from this state is up to the application on the Psion Teklogix computer using the remote sockets API.
- Synchronization between the network controller and the Psion Teklogix computer.

  When a Psion Teklogix computer goes out of range of a base station, socket calls are queued / blocked until the computer is back in range. If the socket state changes during this time, it is the responsibility of the application on the Psion Teklogix computer to recover from this state.

  If a Psion Teklogix computer is powered down, or reset, it may not receive the same terminal number previously assigned to it when it is powered up again (this behaviour is implicitly related to how the socket is created—as each socket is the equivalent to a session in the Psion Teklogix computer, and hence appears as a separate terminal to the network controller software). This behaviour has several consequences, one of which is that a socket may remain open on the controller for significant periods of time without a corresponding Psion Teklogix computer session (a stale connection).
- Terminal number assignment.

  Terminal numbers are assigned dynamically by the network controller when a socket is opened (one session with one assigned terminal number for each socket). As a result, it may be difficult to determine from a Psion Teklogix computer the terminal number(s) assigned to

it. Also, as mentioned above, there is no guarantee a Psion Teklogix computer that closes and then re-opens a socket will be reassigned the same terminal number by the network controller.

- Remote socket calls will not always generate errors in the same way as Windows Socket 2 calls.

  Due to the nature of the underlying protocol, socket calls (**send**, for example) will not immediately return errors to the application. If a socket call generates an error, it will usually be the next call (or some future call) that actually reports the error to the application. This behaviour should not have any obvious consequences, unless the application makes other assumptions; e.g., that the successful completion of a **send** call guarantees delivery of a packet to the partner device (host). However, this behaviour is present in TCP/IP, if in a less obvious way. Applications needing to guarantee arrival of a message must wait for an acknowledgement from the host in order to confirm successful delivery of the data.

- Available Bandwidth.

  As with any wireless device, the amount of bandwidth available to it will generally be lower than that available to a wired device due to the overhead of the radio protocols, possible interference, etc. Applications should restrict the amount of data sent to the Psion Teklogix computer (especially over narrowband and similar restricted-bandwidth media) to prevent poor response times.

- Binding to an IP address/port.

  The bind function on the Psion Teklogix computer behaves in a slightly unusual fashion due to the nature of remote sockets. The bind call causes the controller, rather than the Psion Teklogix computer, to bind a particular IP address and port. As a result, every Psion Teklogix computer making a bind call must specify either: (a) a unique port number, or (b) a unique IP address for the bind. Failure to do so will result in one Psion Teklogix computer either failing to successfully bind a port, or failing to successfully correct / accept connections. Psion Teklogix computer IP addresses cannot be assigned via bootp or DHCP as the Psion Teklogix computers with narrowband radios may not have MAC addresses.

- Number of remote sockets available is limited.

  The following restrictions apply to the number of remote sockets that may be in use:

  - A maximum of seven remote sockets can be open for the same Psion Teklogix computer.
  - The maximum number of remote sockets that can be open on the network controller is defined in the controller configuration. See “Configuring A Psion Teklogix Network Controller (9400/9500)” on page 80.

- Address matching on the network controller.

  When a Psion Teklogix computer sends a socket bind request with a non-zero IP address as a parameter, the network controller attempts to create and use that IP address as an alias IP address. This is an alternate IP address that the controller will send/receive packets from, in addition to its own native IP address. To create the alias the controller chooses an interface with maximum number of matches between alias IP address and the interface IP addresses. It makes binary comparison of the addresses. Any match with fewer than 16 identical bits is ignored, as it would be hard for a host to route to such an alias.

  Here is an example of address matching:

  10.128.0.154 = 0x0a80009a = 00001010 10000000 00000000 10011010
  10.128.78.19 = 0x0a804e13 = 00001010 10000000 01001110 00010011
  |_____________________|
  |
  17 (bit) matches

13.3.7 Remote Sockets API Elements

C++: Remote sockets on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix::RemoteSockets** namespace.

Java: Not available.

.NET: Not available.

13.4 Technical References

This chapter assumes that the reader has some familiarity with Sockets programming. The following books by W. Richard Stevens are recommended references to Sockets programming:

- Unix Network Programming - Networking APIs: Sockets and XTI, W.Richard Stevens, volume 1, 2nd edition. Prentice Hall, 1998, ISBN 0-13-490012-X.
- TCP/IP Illustrated, Volume 1 The Protocols, W. Richard Stevens, Addison-Wesley, 1995, ISBN 0-201-63346-9.
- TCP/IP Illustrated, Volume 2 The Implementation, Gary R. Wright and W. Richard Stevens. Addison-Wesley, 1995, ISBN 0-201-63354-X.

SCANNERS 14

14.1 Overview 85
14.1.1 Types Of Scanners 85
14.1.1.1 Non-decoded Scanners. 85
14.1.1.2 Decoded Scanners 85
14.1.1.3 Imaging Scanners 85
14.1.2 Scanner Connections 85
14.1.2.1 Internal Scanners 86
14.1.2.2 External Scanners 86
14.1.3 Symbologies 87
14.2 Configuring Scanners 88
14.2.1 Configuring Through The Windows Control Panel 88
14.2.2 Configuring Through The SDK. 89
14.2.3 Configuring With Bar Codes 89
14.3 Configuring Scanners Through The Mobile Devices SDK. 89
14.3.1 Configuring Scanner Properties. 90
14.3.1.1 Double-click Settings 90
14.3.1.2 Display, Beep, And Logging Options 91
14.3.2 Configuring Global Bar Code Options 92
14.3.2.1 Non-decoded Scanner Options 92
14.3.2.2 Decoded Scanner Options 92
14.3.2.3 Decoded Scanner Advanced Options 93
14.3.2.4 Decoded Scanner 2D Options 93
14.3.2.5 Decoded Scanner Data Options 94
14.3.2.6 Imager Options (Deprecated) 94
14.3.2.7 Imager Advanced Options (Deprecated) 95
14.3.3 Configuring Bar Code Symbologies 96
14.3.3.1 Bar Code Field Settings 96
14.3.3.2 Code 39 Settings 97
14.3.3.3 Trioptic Code Settings 100
14.3.3.4 Code 128 Settings 101
14.3.3.5 EAN 13 Settings 103
14.3.3.6 EAN 8 Settings. 105
14.3.3.7 UPC A Settings 108
14.3.3.8 UPC E Settings 110
14.3.3.9 UPC/EAN Shared Settings 113
14.3.3.10 Codabar Settings 114
14.3.3.11 Code 93 Settings 116
14.3.3.12 Code 11 Settings 119
14.3.3.13 Interleaved 2 of 5 Settings 120
14.3.3.14 MSI Plessey Settings 123
14.3.3.15 Matrix 2 of 5 Settings. 125
14.3.3.16 Discrete 2 of 5 Settings 126
14.3.3.17 IATA 2 of 5 Settings 128
14.3.3.18 Telepen Settings 129
14.3.3.19 RSS Code Settings 130
14.3.3.20 PosiCode Settings 132

14.3.3.21 Composite Settings 132
14.3.3.22 TLC-39 Settings 134
14.3.3.23 PDF417 Settings 135
14.3.3.24 Micro PDF-417 Settings 137
14.3.3.25 Code 16K Settings 138
14.3.3.26 Code 49 Settings 139
14.3.3.27 Codablock Settings 139
14.3.3.28 2D Data Matrix Settings 140
14.3.3.29 2D QR Code Settings. 141
14.3.3.30 2D MaxiCode Settings 142
14.3.3.31 2D Aztec Settings. 143
14.3.3.32 Postal: PlaNET Settings 144
14.3.3.33 Postal: PostNET Settings 145
14.3.3.34 Postal: Australian Settings 146
14.3.3.35 Postal: Canadian Settings. 146
14.3.3.36 Postal: Japanese Settings 147
14.3.3.37 Postal: Kix Settings. 148
14.3.3.38 Postal: Korean Settings 149
14.3.3.39 Postal: Royal Settings 150
14.4 Reading Bar Codes 150
14.4.1 Initiating A Bar Code Scan 150
14.4.2 Controlling A Bar Code Scan 151
14.5 Scanner Events 152

14.1 Overview

The Mobile Devices SDK provides functions that handle all the low-level activities of bar code scanners.

14.1.1 Types Of Scanners

The Mobile Devices SDK supports the following types of bar code scanner:

- Non-decoded scanners
- Decoded scanners
- Imaging scanners (for legacy applications only, see Section 14.1.1.3 below)

14.1.1.1 Non-decoded Scanners

Non-decoded scanners present unprocessed scan data to the computer. These scanners rely on the computer to do the decoding. The Mobile Devices SDK returns the scan data as a character string to the calling program.

The non-decoded scan driver receives scan data from the peripheral controller and decodes it. It passes it to the Scan Control Service, which performs any necessary translations and delivers the data, through the SDK, as a character string to the application.

14.1.1.2 Decoded Scanners

Decoded scanners have built in decoders; they process the raw scan data themselves. The formatted data is passed to the scanner driver; then to the Scan Control Service; then to the SDK; and finally to the application. Both 1D and 2D decoded laser scanners are supported by the Mobile Devices SDK. The data is received as an undelimited character string.

14.1.1.3 Imaging Scanners

The Mobile Devices SDK supports bar code scanning by legacy imagers; however, the Mobile Devices SDK should not be used for new applications involving imagers.

Important: ***New applications must use the Mobile Devices SDK for laser bar code scanners only. The Imaging Services SDK must be used for all imaging scanner applications including reading bar codes.***

An imaging scanner can take digital photographs in addition to scanning bar codes. The default mode for an imaging scanner is to act as a decoded scanner. The Mobile Devices SDK provides methods for putting an internal imaging scanner into image capture mode and returning it to bar code scanning mode, as well as for controlling it in image capture mode. The SDK does not control an externally connected imaging scanner when it is in imaging mode.

14.1.2 Scanner Connections

Scanners are connected to the computer in one of the following ways:

- Internal
- External through the tether port
- External through a vehicle-mount cradle

14.1.2.1 Internal Scanners

An internal scanner is built into the body of the hand-held computer. Each hand-held computer can have only one internal scanner. The scanner can be a non-decoded scanner, a decoded scanner, an imaging scanner, or an RFID scanner. The internal scanner is activated by the hand-held's scanner button, or by the pistol grip trigger mechanism.

Pressing the scan button on the keyboard and pressing the trigger on the pistol grip have identical effects. In the rest of this description this will be referred to as pressing the trigger on the hand-held computer.

Querying the scanner type of an internal scanner through the API returns one of the following strings:

Scanner Type	String Returned By API
Non-decoded scanner	Symbol 1200 [HP | LR | ALR | WA]
Decoded scanner	Intermec E1022
	Intermec EV15
	RfidSirit Oem186 (RFID/1d decoded)
	Symbol 955
	Symbol 1223 [HP | LR | ALR | WA]
	Symbol 1224 HP
	Symbol 1524 ER
	Symbol 2223
	Symbol 923 HP
Imaging scanner	HHP IT4080
	Symagery IL6303
	Symagery SX4000 [ST | ULR | UHD]
	Symagery SX5303 [HD | IL | ST | ULR | UHD]
	Symagery SX5393
	Symagery SX5400
Unidentified scanner	Unknown
	Not configured
	No Internal
	None

14.1.2.2 External Scanners

A Psion Teklogix computer can have up to two external scanners. Only one of them can be a non-decoded scanner. The scanners can be connected through the tether port, or through a vehicle-mount cradle.

Querying the scanner type for an external scanner through the Mobile Devices SDK returns one of the following strings:

Scanner Type	String Returned By API
No external scanner	No external scanner
Non-decoded scanner	Non-decoded scanner
Decoded scanner	Serial scanner

14.1.3 Symbologies

In bar codes, numbers, letters, and computer-recognized characters are represented in a combination of bars and spaces. The rules for encoding the characters in alternating bars and spaces of varying widths are called a symbology. There are many standard symbologies. Each is preferred for certain types of applications.

The following symbologies are decoded on Psion Teklogix Windows-based computers:

Symbology	Non-decoded	Imager (Deprecated)	Decoded Scanner			
			Symbol 1223 Symbol 923	Symbol 1224	Symbol 1524	Symbol 2223
Code 39	Yes	Yes	Yes	Yes	Yes	Yes
Code 128	Yes	Yes	Yes	Yes	Yes	Yes
UPC/EAN	Yes	Yes	No	No	No	Yes
EAN 13	Yes	Yes	Yes	Yes	Yes	Yes
EAN 8	Yes	Yes	Yes	Yes	Yes	Yes
UPC A	Yes	Yes	Yes	Yes	Yes	Yes
UPC E	Yes	Yes	Yes	Yes	Yes	Yes
Codabar	Yes	Yes	Yes	Yes	Yes	Yes
Code 93	Yes	Yes	Yes	Yes	Yes	Yes
Code 11	Yes	No	No	No	No	No
Interleaved 2 of 5	Yes	Yes	Yes	Yes	Yes	Yes
MSI Plessey	Yes	No	Yes	Yes	Yes	Yes
Discrete 2 of 5	Yes	No	Yes	Yes	Yes	Yes
IATA 2 of 5	Yes	No	Yes	No	No	Yes
Australia Post Address Code	No	Yes	No	No	No	No
Japanese Postal Code	No	Yes	No	No	No	No
Korean Postal Code	No	Yes	No	No	No	No

Symbology	Non-decoded	Imager (Deprecated)	Decoded Scanner			
			Symbol 1223 Symbol 923	Symbol 1224	Symbol 1524	Symbol 2223
Royal Mail Code	No	Yes	No	No	No	No
PLANET	No	Yes	No	No	No	No
POSTNET	No	Yes	No	No	No	No
2D Data Matrix	No	Yes	No	No	No	No
2D MaxiCode	No	Yes	No	No	No	No
2D PDF417	No	Yes	No	No	No	Yes
2D Micro PDF417	No	Yes	No	No	No	Yes
2D QR Code	No	Yes	No	No	No	No
RSS	No	Yes	No	Yes	Yes	Yes
Aztec Code	No	Yes	No	No	No	No
Composite Codes	No	Yes	No	No	No	Yes

14.2 Configuring Scanners

Scanners are configured using the following methods:

- Through the Windows Control Panel Scanner Properties
- Through application software using the SDK
- By scanning configuration bar codes

All Windows Control Panel settings can also be configured through the SDK. The most recently set value, from either source, applies. Some scanner parameters, for example those controlling an imaging scanner in image capture mode, can only be set through the SDK.

When parameters are set through the Control Panel or through the SDK, all accessible parameters can be set. It is possible to set values for parameters that are not relevant to the scanner connected to the Psion Teklogix computer. These settings are ignored at runtime.

In some situations, the internal and external scanners share the same configuration settings. All scanners using the same symbology share the same field settings. For example, if there is an internal non-decoded scanner and an external non-decoded scanner, they will both use the same configuration settings.

Only external decoded scanners and RFID scanners are configured by scanning configuration bar codes, provided by the scanner manufacturer.

14.2.1 Configuring Through The Windows Control Panel

Scanner settings for bar codes are configured through the Windows Control Panel. Psion Teklogix Windows-based computers have a **Teklogix Scanners** applet in the Control Panel. Selecting this icon opens the **Scanner Properties** window. Refer to the user manual for the Psion Teklogix computer for instructions.

14.2.2 Configuring Through The SDK

All the settings that can be configured through the Windows Control Panel can be configured through the Mobile Devices SDK with the exception of the translation features.

14.2.3 Configuring With Bar Codes

All external decoded scanners must be configured by scanning special purpose bar codes supplied by the scanner manufacturer. For a list of the relevant programming manuals from the scanner manufacturers refer to the user manual for the Psion Teklogix computer. Scanners which must be configured using bar codes cannot be configured through the Mobile Devices SDK.

For the bar codes needed to configure the Symbol Technologies decoded scanners used with Psion Teklogix computers refer to “Chapter 11, Parameter Menus”, in MiniScan MS XX04 Series Integration Reference Guide (Part number 72-67134-02), published by Symbol Technologies. This manual can be downloaded from the Symbol Technologies site at:

http://support.symbol.com/support/product/products.do?id=m4 (under the “OEM Products” Product Line, Product Number “MSxx04”).

14.3 Configuring Scanners Through The Mobile Devices SDK

This section describes the scanner settings that are available through the Mobile Devices SDK. Each setting is listed with its Control Panel equivalent. The effects of the settings are not described here as complete details are contained in the user manual for each Psion Teklogix computer. The user manuals also list the internal scanner models that are available for each hand-held computer, with the symbologies supported by each scanner.

Each of the API libraries has methods for the following (refer to the documentation for the relevant API library for details):

Method Purpose	Input & Output
Set a scanner setting	Input: String, Object <scanner setting key>\<scanner setting name>, <value>
Get a scanner setting	Input: String <scanner setting key>\<scanner setting name> Output: Object <scanner setting value>
Get a scanner setting name	Input: Integer <scanner setting index> Output: String <scanner setting key>\<scanner setting name>

<**scanner setting key**> is a string identifying a group of related scanner settings

<**scanner setting name**> is a string identifying a scanner setting

<**scanner setting key**>\<**scanner setting name**> together uniquely identify a scanner setting
For example:
Scs\Scanresult
Barcode\C39\Decoded\Check Digit Verification

<**scanner setting value**> is an object (usually integer) representing the current value of the setting

<**scanner setting index**> is a unique integer that identifies each <**scanner setting key**>\<**scanner setting name**> combination. This integer is not the same for all versions of the Mobile Devices SDK.

The following .NET code sample illustrates how to get and set scanner variables using the setting key/name methods, and also illustrates how to create a list of all scanner setting index and key/name pairs.

```
// Create an instance of a ScannerServicesDriver object
PsionTeklogix.Barcode.ScannerServices.ScannerServicesDriver myScanner =
    new PsionTeklogix.Barcode.ScannerServices.ScannerServicesDriver();

private void btnGetInfo_Click(object sender, EventArgs e)
{
    // Get Click Data and Click Time values
    // Note double backslashes due to escape sequence
    tbClickData.Text = myScanner.GetProperty("Scs\\Click Data").ToString();
    tbClickTime.Text = myScanner.GetProperty("Scs\\Click Time").ToString();

    // Cycle through all scanner settings and list the index numbers
    // and setting names in a combo box
    for (int i = 0; i < myScanner.TotalSettingsCount; i++)
       cbScanSetting.Items.Add(i.ToString() + ": "
        + myScanner.GetSettingName(i));
}

private void btnSetInfo_Click(object sender, EventArgs e)
{
    // Set Click Data and Click Time values
    myScanner.SetProperty("Scs\\Click Data", tbClickData.Text);
    myScanner.SetProperty("Scs\\Click Time", int.Parse(tbClickTime.Text));
}
```

14.3.1 Configuring Scanner Properties

The following scanner behaviours can be configured using the Mobile Devices SDK:

- Double-click settings
- Display settings
- Beep settings
- Logging settings

14.3.1.1 Double-click Settings

A keyboard key and/or a grip trigger can be registered as the trigger for an internal scanner. See "Keyboard Remapping" on page 56 for information on how to do this. The double-click time for the trigger associated with the scanner is set as described in this section.

A double-click occurs when the scanner-associated trigger is pressed twice within the period defined in the **Click Time** setting. The trigger does not have to be released after the second press within the **Click Time** in order for the double-click to register.

Double-clicking has two modes. A method exists for toggling the double-click between these modes. The modes are as follows:

- Scan a bar code.
- Send a character to the application.

Scan a bar code mode

In this mode a bar code is scanned in exactly the same way as when the scanner-associated trigger is pressed once. This is the default setting.

Send a character to the application mode

In this mode, double-clicking the scanner-associated trigger sends a single pre-selected character defined in the **Click Data** setting to the application.

These settings control this process for the internal scanner as well as for an external non-decoded scanner.

Control Panel:

These settings can be changed through the Control Panel Scanner Properties as follows:

- Tab Options
- Menu Item Double Click

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Scs				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Click Time	0 ms	1000 ms	250 ms	Click Time
Click Data	0	0xFFFF	0	Click Data

14.3.1.2 Display, Beep, And Logging Options

These settings control how bar codes are displayed on the screen, when beeps are generated, and whether bar code scans are logged. These settings are applied to the internal scanner as well as an external non-decoded scanner.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Options
- Menu Item Display

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Scs				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Scan Result	0 = off	1 = on	1	Scan Result
Scan Indic	0 = off	1 = on	1	Scan Indicator
Result Time	0 s	20 s	0 s	Scan Result Time
Scan Good Beep	0 = off	1 = on	1	Good Scan Beep
Scan Failed Beep	0 = off	1 = on	1	Bad Scan Beep
Scan Log File	0 = off	1 = on	0	Scan Log File
Soft Scan Timeout	1 s	10 s	3 s	Soft Scan Timeout

14.3.2 Configuring Global Bar Code Options

14.3.2.1 Non-decoded Scanner Options

These settings apply to internal and external non-decoded scanners.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Options

SDK

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: NonDecoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Dot Time	0 ms	3000 ms	0 ms	Dot Time
Short Code	0 = off	1 = on	0	Short Code
Verify	0	15	0	Verify
Security	0	99	30	Security

14.3.2.2 Decoded Scanner Options

These settings apply to the internal decoded scanner only.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Options

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Aiming Mode	0	2	0	Dot Time
Aim Duration	0	30	0	Aim Duration
Laser On Time	5	99	50	Laser On Time

14.3.2.3 Decoded Scanner Advanced Options

These settings apply to internal decoded scanners (including imagers acting as bar code scanners) only.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Advanced Options

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Continuous Scan	0 = off	1 = on	0	Continuous Scan Mode
Minimum Cancel Time	0	500	0	Minimum Cancel Time
Power Mode	0	1	1	Power Mode
Time Delay To Low Power	0	3	0	Low Power Timeout
Parameter Scanning	0 = off	1 = on	1	Parameter Scanning
Linear Code Type Security Levels	1	4	2	Linear Security Level
Bi-directional Redundancy	0 = off	1 = on	0	Bi-directional Redundancy
Scan Angle	181	182	182	Scan Angle
Scanning Mode	1	7	1	Scanning Mode
Raster Height	1	15	15	Raster Height
Raster Expansion Rate	1	15	11	Raster Expand Rate

14.3.2.4 Decoded Scanner 2D Options

These settings apply to the internal decoded scanner only.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item 2D Scanning Options

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Scanning Mode	1	7	1	Scanning Mode
Raster Height	1	15	15	Raster Height
Raster Expansion Rate	1	15	11	Raster Expand Rate

14.3.2.5 Decoded Scanner Data Options

These settings apply to the internal decoded scanner only.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Data Options

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Transmit Code ID Character	0	2	0	Transmit Code ID Char
Scan Data Transmission Format	0	7	0	Scan Data Format
Prefix	0	0xFF	0	Prefix <P>
Suffix 1	0	0xFF	13	Suffix <S1>
Suffix 2	0	0xFF	10	Suffix <S2>
Delete Character Set ECIs	0 = off	1 = on	0	Delete Char Set ECIs
ECI Decoder	0 = off	1 = on	0	ECI Decoder

14.3.2.6 Imager Options (Deprecated)

Important: ***New applications must use the Mobile Devices SDK for laser bar code scanners only. The Imaging Services SDK must be used for all imaging scanner applications including reading bar codes.***

These settings apply to the internal imaging scanner only. An imaging scanner captures a bar code as an image, then decodes the bar code from that image. The settings listed here only apply to the imager in bar code scanning mode. There is a separate set of configuration parameters for the image capture mode—these can only be set through the SDK.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Options

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
TekImager Enabled	0 = off	1 = on	1	TekImager Enabled
ContinuousScanMode	0 = off	1 = on	0	Continuous Scan Mode
OnlyDecodeCenter	0 = off	1 = on	0	Center Barcode Only
MaxNumberBarcodes	1	6	1	Max Number Barcodes
MinNumberBarcodes	1	6	1	Barcodes Must Decode
WindowWidth	128	1280	900	Window Width
WindowHeight	128	1024	500	Window Height
Dot Time	0 ms	3000 ms	0 ms	Dot Time
AutoExposure	0 = off	1 = on	1	Auto Exposure
FastConverge	0 = off	1 = on	0	Fast Converge
MaxGain	357	7920	7680	Max Gain
MaxIntegration	0	0xFFFF	26170	Max Integration
MaxIllumination	0	7	7	Max Illumination

14.3.2.7 Imager Advanced Options (Deprecated)

Important: ***New applications must use the Mobile Devices SDK for laser bar code scanners only. The Imaging Services SDK must be used for all imaging scanner applications including reading bar codes***

These settings apply to the internal imaging scanner only. An imaging scanner captures a bar code as an image, then decodes the bar code from that image. The settings listed here only apply to the imager in bar code scanning mode. There is a separate set of configuration parameters for the image capture mode—these can only be set through the SDK.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Advanced Options

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
DefaultDevice	0 = off	1 = on	0	Factory Defaults on Reboot
MinScanDuration	0	6	3	Min Scan Duration
MaxCapturesPerTrigger	1	32	9	Captures Per HW Trigger
AutoExposure	0 = off	1 = on	1	Auto Exposure
FastConverge	0 = off	1 = on	0	Fast Converge
MaxGain	2	30	30	Max Gain
MaxIntegration	0	43	18	Max Integration
MaxIllumination	0	7	7	Max Illumination
DecoderTimeout	200	800	500	Decoder Timeout
AdaptiveWindowing	0 = off	1 = on	0	Adaptive Windowing
ConstantIllumination	0 = off	1 = on	0	Constant Illumination

14.3.3 Configuring Bar Code Symbologies

14.3.3.1 Bar Code Field Settings

Field Settings (All Scanners)

These field settings are used to control length filters and data stripping parameters. They apply to nearly all bar code symbologies and scanners.

Control Panel:

These field settings may be changed in each bar code symbology submenu on the non-decoded, decoded, or imager menus; however, there is only one set of data fields for each symbology. Changing these settings for one scanner type changes the settings for the same symbology all scanner types. These settings are changed through the Control Panel Scanner Properties as follows:

- Tab — Barcodes
- Scanner: — All
- Menu Item — [Bar Code Symbology]→ Length Restriction
  [Bar Code Symbology] → Add/Remove Data

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\[Symbology Code][1]\Scs				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Field Size	0	1400	0	Field Size
Minimum Size	0	1400	0	Minimum Size
Maximum Size	0	1400	0	Maximum Size
Prefix Char	0	0xFFFF	0	Prefix Char
Suffix Char	0	0xFFFF	0	Suffix Char
Strip Leading	0	127	0	Strip Leading
Strip Trailing	0	127	0	Strip Trailing
[1]Replace "[Symbology Code]" with the code found in the appropriate symbology section below (e.g. "C39", "I25", "Codabar", etc.)				

14.3.3.2 Code 39 Settings

Non-decoded Scanner

These settings apply to an internal non-decoded, and an external non-decoded, scanner using the Code 39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Code 39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C39				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Full Ascii	0 = off	1 = on	0	Full ASCII
Include Chk	0 = off	1 = on	0	Include Check
AIAG Strip	0 = off	1 = on	0	AIAG Strip
Err Accept	0 = off	1 = on	0	Error Accept
Mod Chk Base	0	2	0	Mod Checks
Transmit Code ID				Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the Code 39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Code 39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C39\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Trioptic	0 = off	1 = on	0	Enable Trioptic Code 39
Convert to Code 32	0 = off	1 = on	0	Convert to Code 32
Code 32 Prefix	0 = off	1 = on	1	Code 32 Prefix
Length L1	0	55	1	Set Length L1
Length L2	0	55	55	Set Length L2
Check Digit Verification	0 = off	1 = on	0	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Full Ascii Conversion	0 = off	1 = on	0	Full ASCII
Decode Performance	0 = off	1 = on	1	Decode Performance
Decode Performance Level	1	3	1	Decode Perf. Level

Imager (Deprecated)

These settings apply to an internal imager scanner using the Code 39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Code 39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C39\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Full ASCII	0 = off	1 = on	0	Full ASCII
Check Digit Verification	0 = off	1 = on	0	Check Digit Verification
Include Check	0 = off	1 = on	0	Include Check

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Code 39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Code 39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C39\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Full Ascii Conversion	0 = off	1 = on	0	Full ASCII
Reading Range	0	1	1	Reading Range
Start Stop Transmission	0 = off	1 = on	0	Start/Stop Transmit
Accepted Start Character	1	3	3	Accepted Start Char
Check Digit Verification	0	3	0	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Length L1	0	255	0	Minimum Length

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Code 39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Code 39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C39\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Start Stop Char	0 = off	1 = on	0	Start/Stop Transmit
Check Char	0	2	0	Check Char
Length Min	0	48	0	Minimum Length
Length Max	0	48	48	Maximum Length
Append	0 = off	1 = on	0	Append
Pharmaceutical	0 = off	1 = on	0	Pharmaceutical
Full Ascii	0 = off	1 = on	0	Full ASCII

14.3.3.3 Trioptic Code Settings

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Trioptic Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Trioptic Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Trioptic\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.3.3.4 Code 128 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the Code 128 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Code 128

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C128				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Include Sym	0 = off	1 = on	1	Include Sym
Variant Mode	0	3	1	Variations
Transmit Code ID Char	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the Code 128 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Code 128

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C128\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
UCC EAN 128	0 = off	1 = on	1	Enable GS1-128/GS1 US
ISBT 128	0 = off	1 = on	0	Enable ISBT 128
Decode Performance	0 = off	1 = on	1	Decode Performance
Decode Performance Level	1	3	1	Decode Perf. Level

Imager (Deprecated)

These settings apply to an internal imager scanner using the Code 128 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Code 128

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C128\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Code 128 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Code 128

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C128\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
EAN 128	0 = off	1 = on	1	GS1-128
EAN 128 Identifier	0 = off	1 = on	1	GS1-128 Identifier
GTIN Compliant	0 = off	1 = on	0	GTIN Compliant
FNC1 Conversion	0	255	29	FNC1 Conversion
ISBT 128	0 = off	1 = on	0	Enable ISBT128
ISBT Concatenation Transmission	0	2	0	ISBT Concat Transmit
ISBT Concatenate Pair	0 = off	1 = on	0	ISBT Concat Any Pair
Reading Range	0	1	1	Reading Range
Check Digit Verification	0	1	0	Check Digit Verification
Length L1	0	255	0	Minimum Length

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Code 128symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Code 128

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C128\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
ISBT Concatenation	0 = off	1 = on	0	ISBT Concatenation
Length Min	0	80	0	Minimum Length
Length Max	0	80	80	Maximum Length

14.3.3.5 EAN 13 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the EAN 13 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item EAN 13

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN13				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Enable Bookland	0 = off	1 = on	0	Enable Bookland EAN
Inc Country	0 = off	1 = on	1	Include Country
Include Chk	0 = off	1 = on	1	Include Check
Addendum	0	2	0	Addendum
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the EAN 13 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item EAN 13

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN13\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled

Imager (Deprecated)

These settings apply to an internal imager scanner using the EAN 13 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item EAN 13

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN13\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Addendum	0	2	0	Addendum

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the EAN 13 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item EAN 13

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN13\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
ISBN Conversion	0 = off	1 = on	0	ISBN Conversion
Transmit Check Digit	0 = off	1 = on	1	Transmit Check Digit

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the EAN 13 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item EAN 13

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN13\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Check Digit	0 = off	1 = on	1	Check Digit
2 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 2
5 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 5
Addendum Required	0 = off	1 = on	0	Addendum Required
Addendum Separator	0 = off	1 = on	1	Addendum Separator
ISBN Translate	0 = off	1 = on	0	ISBN Translate

14.3.3.6 EAN 8 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the EAN 8 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item EAN 8

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN8				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Include Chk	0 = off	1 = on	1	Include Check
Addendum	0	2	0	Addendum
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the EAN 8 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item EAN 8

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN8\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
EAN8 Zero Extend	0 = off	1 = on	0	EAN-8 Zero Extend

Imager (Deprecated)

These settings apply to an internal imager scanner using the EAN 8 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item EAN 8

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN8\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Addendum	0	2	0	Addendum

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the EAN 8 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item EAN 8

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN8\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Transmit Check Digit	0 = off	1 = on	1	Transmit Check Digit
Transmit as EAN-13	0 = off	1 = on	0	Convert to EAN 13

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the EAN 8 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item EAN 8

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\EAN8\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Check Digit	0 = off	1 = on	1	Check Digit
2 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 2
5 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 5
Addendum Required	0 = off	1 = on	0	Addendum Required
Addendum Separator	0 = off	1 = on	1	Addendum Separator

14.3.3.7 UPC A Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the UPC A symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item UPC A

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCA				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Inc Num Sys	0 = off	1 = on	0	Include Number Sys
Include Chk	0 = off	1 = on	0	Include Check
Addendum	0	2	0	Addendum
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the UPC A symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item UPC A

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCA\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
UPCA Transmit Check Digit	0 = off	1 = on	1	UPC-A Check Digit
UPCA Preamble	0	2	1	UPC-A Preamble

Imager (Deprecated)

These settings apply to an internal imager scanner using the UPC A symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item UPC A

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCA\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Addendum	0	2	0	Addendum

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the UPC A symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item UPC A

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCA\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Transmit Check Digit	0 = off	1 = on	1	Transmit Check Digit
Transmit Number System	0 = off	1 = on	1	Transmit Number System
Transmit as EAN-13	0 = off	1 = on	0	Convert to EAN 13

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the UPC A symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item UPC A

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCA\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Check Digit	0 = off	1 = on	1	Check Digit
Number System	0 = off	1 = on	1	Transmit Number System
2 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 2
5 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 5
Addendum Required	0 = off	1 = on	0	Addendum Required
Addendum Separator	0 = off	1 = on	1	Addendum Separator

14.3.3.8 UPC E Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the UPC E symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item UPC E

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCE				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Exp to UPC A	0 = off	1 = on	1	Convert to UPC-A
Inc Num Sys	0 = off	1 = on	1	Include Number Sys
Include Chk	0 = off	1 = on	1	Include Check
Addendum	0	2	0	Addendum
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Imager (Deprecated)

These settings apply to an internal imager scanner using the UPC E symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item UPC E

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCE\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Addendum	0	2	0	Addendum

Decoded Scanner

These settings apply to an internal decoded scanner using the UPC E symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item UPC E

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCE\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
UPCE Enabled	0 = off	1 = on	1	Enable UPC-E
UPCE1 Enabled	0 = off	1 = on	0	Enable UPC-E1
UPCE Transmit Check Digit	0 = off	1 = on	1	UPC-E Check Digit
UPCE1 Transmit Check Digit	0 = off	1 = on	1	UPC-E1 Check Digit
UPCE Preamble	0	2	1	UPC-E Preamble
UPCE1 Preamble	0	2	1	UPC-E1 Preamble
Convert UPCE to UPCA	0 = off	1 = on	0	Conv. UPC-E to UPC-A
Convert UPCE1 to UPCA	0 = off	1 = on	0	Conv. UPC-E1 to UPC-A

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the UPC E symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item UPC E

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCE\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
UPC-E1 Enabled	0 = off	1 = on	0	Enable UPC-E1
Transmit Check Digit	0 = off	1 = on	1	Transmit Check Digit
Transmit Number System	0 = off	1 = on	1	Transmit Number System
Transmit as UPC-A	0 = off	1 = on	0	Convert to UPC-A

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the UPC E symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item UPC E

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPCE\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
UPC-E1 Enabled	0 = off	1 = on	0	Enable UPC-E1
Expand	0 = off	1 = on	0	Expand
Check Digit	0 = off	1 = on	1	Check Digit
Number System	0 = off	1 = on	1	Transmit Number System
2 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 2
5 Digit Addendum	0 = off	1 = on	0	Addendum Add-on 5
Addendum Required	0 = off	1 = on	0	Addendum Required
Addendum Separator	0 = off	1 = on	1	Addendum Separator

14.3.3.9 UPC/EAN Shared Settings

Decoded Scanner

These settings apply to an internal decoded scanner using UPC/EAN symbologies.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item UPC/EAN Shared Settings

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPC_EAN\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Bookland Enabled	0 = off	1 = on	0	Enable Bookland EAN
Supplementals	0	2	0	Supplementals
Supplemental Redundancy	2	20	20	Supp. Redundancy
Security Level	0	3	0	Security Level
Linear Decode	0 = off	1 = on	0	Linear Decode
UPC Half Block Stitching	0 = off	1 = on	1	UPC Half Block Stitching

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using UPC/EAN symbologies.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item UPC/EAN Shared Settings

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPC_EAN\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Addendum	0	1	0	Addendum
Addendum Add-on 2	0 = off	1 = on	0	Addendum Add-on 2
Addendum Add-on 5	0 = off	1 = on	0	Addendum Add-on 5
Addendum Security	0	100	10	Addendum Security
GTIN Compliant	0 = off	1 = on	0	GTIN Compliant

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using UPC/EAN symbologies.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item UPC/EAN

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\UPC_EAN\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
UPC-A EAN-13 Extended Coupon Code	0 = off	1 = on	1	Extended Coupon Code

14.3.3.10 Codabar Settings

Decoded Scanner

These settings apply to an internal decoded scanner using the Codabar symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Codabar

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\CDB\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	0	55	5	Set Length L1
Length L2	0	55	55	Set Length L2
CLSI Editing	0 = off	1 = on	0	CLSI Editing
NOTIS Editing	0 = off	1 = on	0	NOTIS Editing

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the Codabar symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Codabar

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\CDB

Scanner Setting Key: Barcode\CDB				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
StartStopChars	0 = off	1 = on	1	Strip Start/Stop Chars
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Imager (Deprecated)

These settings apply to an internal imager scanner using the Codabar symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Codabar

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\CDB\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Codabar symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Codabar

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\CDB\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Start Stop Transmission	0	4	0	Start/Stop Transmit
CLSI Library System	0 = off	1 = on	0	CLSI Library System
Check Digit Verification	0 = off	1 = on	0	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Length L1	0	255	6	Set Length L1
Length L2	0	255	0	Set Length L2
Length L3	0	255	0	Set Length L3
Length Mode	0	1	0	Length Mode

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Codabar symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Codabar

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\CDB\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Start Stop Char	0 = off	1 = on	0	Start/Stop Transmit
Check Char	0	2	0	Check Char
Concatenation	0	2	0	Concatenation
Length Min	2	60	4	Minimum Length
Length Max	2	60	60	Maximum Length

14.3.3.11Code 93 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the Code 93 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Code 93

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C93				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the Code 93 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Code 93

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C93\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	0	55	4	Set Length L1
Length L2	0	55	55	Set Length L2

Imager (Deprecated)

These settings apply to an internal imager scanner using the Code 93 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Code 93

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C93\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Code 93 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Code 93

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C93\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	0	255	1	Set Length L1

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Code 93 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Code 93

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C93\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Length Min	0	80	0	Minimum Length
Length Max	0	80	80	Maximum Length

14.3.3.12Code 11 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the Code 11 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Code 11

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C11				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Include Chk	0 = off	1 = on	0	Include Check
Num Chk Digits	0	2	0	Check Digits
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Code 11 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Code 11

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C11\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Check Digit Verification	1	2	2	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	1	Transmit Check Digit
Length L1	0	255	0	Minimum Length

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Code 11 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Code 11

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C11\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Check Digits Required	0	1	1	Check Digits
Length Min	1	80	4	Minimum Length
Length Max	1	80	80	Maximum Length

14.3.3.13 Interleaved 2 of 5 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the Interleaved 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item Interleaved 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\I25				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Mod Chk Base	0 = off	1 = on	0	MOD 10 Check
ItfChk	0 = off	1 = on	0	ITF Check
Include Chk	0 = off	1 = on	0	Include Check
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the Interleaved 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Interleaved 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\I25\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	0	14	14	Set Length L1
Length L2	0	14	0	Set Length L2
Check Digit Verification	0 = off	1 = on	0	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Convert to EAN 13	0 = off	1 = on	0	Convert to EAN 13

Imager (Deprecated)

These settings apply to an internal imager scanner using the Interleaved 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Interleaved 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\I25\Imager

Scanner Setting Key: Barcode\I25\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Check Digit Verification	0 = off	1 = on	0	Check Digit Verification
Include Check	0 = off	1 = on	0	Include Check

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Interleaved 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Interleaved 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\I25\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Reading Range	0	1	1	Reading Range
Check Digit Verification	0	2	0	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Length L1	0	255	6	Set Length L1
Length L2	0	255	0	Set Length L2
Length L3	0	255	0	Set Length L3
Length Mode	0	1	0	Length Mode

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Interleaved 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Interleaved 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\I25\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Check Digit	0	1	0	Check Digit
Length Min	2	80	4	Minimum Length
Length Max	2	80	80	Maximum Length

14.3.3.14MSI Plessey Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the MSI Plessey symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item MSI Plessey

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MSI				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
One Chk Digits	0 = off	1 = on	1	One Check Digit
Include Chk	0 = off	1 = on	0	Include Check
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the MSI Plessey symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item MSI Plessey

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MSI\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	6	55	6	Set Length L1
Length L2	6	55	55	Set Length L2
Check Digits	0	1	0	Check Digits
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Check Digit Algorithm	0	1	1	Check Digit Algorithm

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the MSI Plessey symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item MSI Plessey

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MSI\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Enabled Plessey	0 = off	1 = on	0	Enable Plessey
Check Digit Verification	1	2	2	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	1	Transmit Check Digit
Plessey Transmit Check Digit	0 = off	1 = on	0	Plessey Transmit Check Digit
Length L1	0	255	6	Minimum Length
Plessey Length L1	0	255	0	Plessey Minimum Length

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the MSI Plessey symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item MSI Plessey

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MSI\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Enabled Plessey	0 = off	1 = on	0	Enable Plessey
Check Char	0	1	0	Check Char
Length Min	4	48	4	Minimum Length
Length Max	4	48	48	Maximum Length
Plessey Length Min	4	48	4	Plessey Minimum Length
Plessey Length Max	4	48	48	Plessey Maximum Length

14.3.3.15 Matrix 2 of 5 Settings

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Matrix 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Matrix 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Matrix25\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	0	255	6	Set Length L1

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Matrix 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Matrix 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Matrix25\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	80	4	Minimum Length
Length Max	1	80	80	Maximum Length

14.3.3.16 Discrete 2 of 5 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the Discrete 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab — Barcodes
- Scanner: — Non-decoded
- Menu Item — Discrete 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\D25				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Mod Chk Base	0 = off	1 = on	0	MOD 10 Check
ItfChk	0 = off	1 = on	0	ITF Check
Include Chk	0 = off	1 = on	0	Include Check
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

Decoded Scanner

These settings apply to an internal decoded scanner using the Discrete 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab — Barcodes
- Scanner: — Decoded (internal)
- Menu Item — Discrete 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\D25\Decoded

Scanner Setting Key: Barcode\D25\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length L1	0	12	12	Set Length L1
Length L2	0	12	0	Set Length L2

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Discrete 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Discrete 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\D25\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Standard 2 of 5 Format	0	1	0	Standard 2 of 5 Format
Check Digit Verification	0	1	0	Check Digit Verification
Transmit Check Digit	0 = off	1 = on	0	Transmit Check Digit
Length L1	0	255	6	Set Length L1
Length L2	0	255	0	Set Length L2
Length L3	0	255	0	Set Length L3
Length Mode	0	1	0	Length Mode

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Discrete 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Discrete 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\D25\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	48	4	Minimum Length
Length Max	1	48	48	Maximum Length

14.3.3.17IATA 2 of 5 Settings

Non-decoded Scanner

These settings apply to an internal or external non-decoded scanner using the IATA 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Non-decoded
- Menu Item IATA 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\IATA25

Scanner Setting Key: Barcode\IATA25				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Mod Chk Base	0 = off	1 = on	0	MOD 10 Check
ItfChk	0 = off	1 = on	0	ITF Check
Include Chk	0 = off	1 = on	0	Include Check
Transmit Code ID Character	0 = off	1 = on	0	Transmit Code ID Char

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the IATA 2 of 5 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item IATA 2 of 5

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\IATA25\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	48	4	Minimum Length
Length Max	1	48	48	Maximum Length

14.3.3.18 Telepen Settings

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Telepen symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Telepen

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Telepen\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Format	0	1	0	Format
Length L1	0	255	0	Set Length L1

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Telepen symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Telepen

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Telepen\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Output	0	1	0	Output
Length Min	1	60	1	Minimum Length
Length Max	1	60	60	Maximum Length

14.3.3.19RSS Code Settings

Decoded Scanner

These settings apply to an internal decoded scanner using the RSS Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item RSS Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\RSSCode\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
RSS-14 Enabled	0 = off	1 = on	0	Enable RSS-14
RSS Limited Enabled	0 = off	1 = on	0	Enable RSS Limited
RSS Expanded Enabled	0 = off	1 = on	0	Enable RSS Expanded

Imager (Deprecated)

These settings apply to an internal imager scanner using the RSS Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item RSS Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\RSSCode\Imager

Scanner Setting Key: Barcode\RSSCode\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the RSS Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item RSS Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\RSSCode\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
RSS-14 Enabled	0 = off	1 = on	0	Enable RSS-14
RSS Limited Enabled	0 = off	1 = on	0	Enable RSS Limited
RSS Expanded Enabled	0 = off	1 = on	0	Enable RSS Expanded

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the RSS Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item RSS Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\RSSCode\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
RSS Limited Enabled	0 = off	1 = on	1	Enable RSS Limited
RSS Expanded Enabled	0 = off	1 = on	1	Enable RSS Expanded
RSS Expanded Length Min	4	74	4	Minimum Length
RSS Expanded Length Max	4	74	74	Maximum Length

14.3.3.20 PosiCode Settings

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the PosiCode symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item PosiCode

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PosiCode\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Posicode	0	2	2	PosiCode
Length Min	2	80	4	Minimum Length
Length Max	2	80	48	Maximum Length

14.3.3.21 Composite Settings

Decoded Scanner

These settings apply to an internal decoded scanner using the Composite Code symbology

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item Composite

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Composite\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
CC-C Enabled	0 = off	1 = on	0	Enable CC-C
CC-AB Enabled	0 = off	1 = on	0	Enable CC-AB
TLC-39 Enabled	0 = off	1 = on	0	Enable TLC-39

Imager (Deprecated)

These settings apply to an internal imager scanner using the Composite Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Composite

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Composite\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Composite symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Composite

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Composite\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
CC-A/B Enabled	0 = off	1 = on	0	Enable CC-AB
CC-C Enabled	0 = off	1 = on	0	Enable CC-C
Linear Transmission only	0 = off	1 = on	0	Linear Transmission only
UPC and EAN composite message decoding	0	2	2	UPC-EAN composite message decoding

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Composite symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Composite

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Composite\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
EAN-UCC Composite Enabled	0 = off	1 = on	0	GS1 128
EAN-UCC Emulation	0 = off	1 = on	0	EAN/UCC 128 Emulation
Length Min	1	2435	1	Minimum Length
Length Max	1	2435	2435	Maximum Length

14.3.3.22TLC-39 Settings

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the TLC-39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item TLC-39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\TLC39\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Linear Transmission only	0 = off	1 = on	0	Linear Transmission only
Security Level	0	100	10	Security Level

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the TLC-39 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item TLC-39

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\TLC39\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.3.3.23PDF417 Settings

Decoded Scanner

These settings apply to an internal decoded scanner using the PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (internal)
- Menu Item PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PDF417\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled

Imager (Deprecated)

These settings apply to an internal imager scanners using the PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PDF417\Imager

Scanner Setting Key: Barcode\PDF417\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PDF417\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PDF417\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Length Min	1	2750	1	Minimum Length
Length Max	1	2750	2750	Maximum Length

14.3.3.24 Micro PDF-417 Settings

Decoded Scanner

These settings apply to an internal decoded scanner using the micro PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab — Barcodes
- Scanner: — Decoded (internal)
- Menu Item — micro PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MicroPDF417\Decoded				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Code 128 Emulation	0 = off	1 = on	0	Code 128 Emulation

Imager (Deprecated)

These settings apply to an internal imager scanner using the micro PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab — Barcodes
- Scanner: — Imager
- Menu Item — micro PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MicroPDF417\Imager

Scanner Setting Key: Barcode\MicroPDF417\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the micro PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item micro PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MicroPDF417\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Code 128 Emulation	0 = off	1 = on	0	Code 128 Emulation

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the micro PDF417 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item micro PDF-417

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MicroPDF417\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	366	1	Minimum Length
Length Max	1	366	366	Maximum Length

14.3.3.25Code 16K Settings

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Code 16K symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Code 16K

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C16K\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	160	1	Minimum Length
Length Max	1	160	160	Maximum Length

14.3.3.26 Code 49 Settings

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Code 49 symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Code 49

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\C49\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	81	1	Minimum Length
Length Max	1	81	81	Maximum Length

14.3.3.27 Codablock Settings

Intermec ISCP Imager

These settings apply only to the Intermec ISCP imager acting as a decoded scanner using the Codablock symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (Intermec ISCP)
- Menu Item Codablock

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Codablock\ICSP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Codablock A Enabled	0 = off	1 = on	0	Enable Codablock A
Codablock F Enabled	0 = off	1 = on	0	Enable Codablock F

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Codablock symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Codablock

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Codablock\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	2048	1	Minimum Length
Length Max	1	2048	2048	Maximum Length

14.3.3.28 2D Data Matrix Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the 2D Data Matrix symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item 2D Data Matrix

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\DataMatrix\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Inverse Video Mode	0 = off	1 = on	0	Inverse Video Mode
Rectangular	0 = off	1 = on	1	Rectangular

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the 2D Data Matrix symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item 2D Data Matrix

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\DataMatrix\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Length Min	1	1500	1	Minimum Length
Length Max	1	1500	1500	Maximum Length

14.3.3.292D QR Code Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the 2D QR Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item 2D QR Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\QRCode\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Inverse Video Mode	0 = off	1 = on	0	Inverse Video Mode

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the 2D QR Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item 2D QR Code

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\QRCode\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Length Min	1	3000	1	Minimum Length
Length Max	1	3000	3000	Maximum Length

14.3.3.302D MaxiCode Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the 2D MaxiCode symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item 2D Maxicode

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MaxiCode\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the 2D MaxiCode symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item 2D Maxicode

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\MaxiCode\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Length Min	1	150	1	Minimum Length
Length Max	1	150	150	Maximum Length

14.3.3.312D Aztec Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the 2D Aztec Code symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item 2D Aztec

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Aztec\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the 2D Aztec symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item 2D Aztec

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\Aztec\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	1	Enabled
Runes Enabled	0 = off	1 = on	0	Aztec Runes
Length Min	1	3750	1	Minimum Length
Length Max	1	3750	3750	Maximum Length

14.3.3.32 Postal: PlaNET Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the PlaNET Postal symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: PlaNET

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PlaNET\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the PlaNET Postal symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: PlaNET

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PlaNET\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Check Digit	0 = off	1 = on	0	Check Digit

14.3.3.33 Postal: PostNET Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the PostNET Postal symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: PostNET

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostNET\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the PostNET Postal symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: PostNET

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostNET\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Check Digit	0 = off	1 = on	0	Check Digit

14.3.3.34 Postal: Australian Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the Australian Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: Australian

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalAus\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Australian Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: Australian

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalAus\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.3.3.35 Postal: Canadian Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the Canadian Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: Canadian

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalCdn\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Canadian Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: Canadian

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalCdn\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.3.3.36 Postal: Japanese Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the Japanese Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: Japanese

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalJap\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Japanese Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: Japanese

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalJap\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.3.3.37 Postal: Kix Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the Netherlands Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: Kix

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalKix\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Netherlands Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: Kix

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalKix\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.3.3.38 Postal: Korean Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the Korean Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: Korean

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalKor\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Korean Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: Korean

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalKor\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled
Length Min	1	80	4	Minimum Length
Length Max	2	80	48	Maximum Length

14.3.3.39 Postal: Royal Settings

Imager (Deprecated)

These settings apply to an internal imager scanner using the Royal Mail Postal symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Imager
- Menu Item Postal: Royal

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalRoyal\Imager				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

HHP Imager

These settings apply only to the HHP imager acting as a decoded scanner using the Royal Mail Postal Address symbology.

Control Panel:

These settings are changed through the Control Panel Scanner Properties as follows:

- Tab Barcodes
- Scanner: Decoded (HHP)
- Menu Item Postal: Royal

SDK:

These settings can be changed and queried through the SDK as follows:

Scanner Setting Key: Barcode\PostalRoyal\HHP				
Scanner Setting Name	Minimum	Maximum	Default	Control Panel Name
Enabled	0 = off	1 = on	0	Enabled

14.4 Reading Bar Codes

14.4.1 Initiating A Bar Code Scan

A bar code scan is initiated for an internal scanner by one of the following:

- Pressing the keyboard key that is configured as the scanner trigger.
- Using the Scan method, in the Scanner API, of the Mobile Devices SDK.

A bar code scan is initiated for an external scanner as follows:

- Pressing the trigger on the pistol grip of the external scanner.

14.4.2 Controlling A Bar Code Scan

The following timing parameters control bar code scanning for the internal scanner:

Double-click time

This parameter defines the time within which the scanner trigger must be pressed twice, if the trigger presses are to be interpreted as a double-click and not as two single-clicks. The following setting controls this for the internal scanner and an external non-decoded scanner:

- **Scs\Click Time:** see "Configuring Scanner Properties" on page 90 for details.

Dot time

For scanners with an aiming dot, this parameter defines the length of time for which the aiming dot is enabled before scanning begins. The following settings control this:

- **Non-decoded\Dot Time;** see "Non-decoded Scanner Options" on page 92 for details.
- **Decoded\Aiming Mode;** see "Decoded Scanner Options" on page 92 for details.
- **Imager\Dot Time;** see "Imager Options (Deprecated)" on page 94 for details.

Scan beam on time

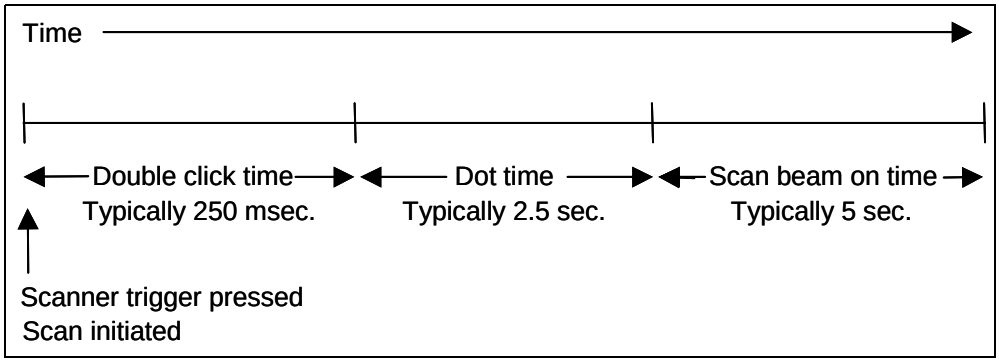
This parameter defines the maximum length of time that the scanning beam is enabled. This can only be set for decoded scanners using the following parameter:

- **Decoded\Laser On Time;** see "Decoded Scanner Options" on page 92 for details.

The only way that the scanning beam can be turned off before the end of this time is by releasing the scanner trigger.

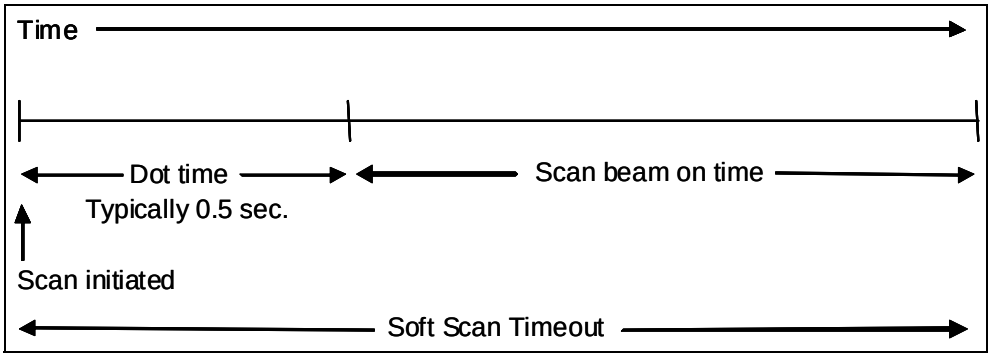
The following diagram shows the timing sequence for an operator controlled bar code scan:

Figure 14.1 Operator Controlled Scanner Timing Sequence



The following diagram shows the timing sequence for a software initiated bar code scan:

Figure 14.2 Software Initiated Scanner Timing Sequence



There are four possible outcomes for a bar code scan:

- The scan is successful.
- The scan times out.
- The scan is cancelled.
- The scan fails.

The scan is successful: The bar code is scanned and decoded before the scanner trigger is released. The decoded bar code is displayed on the screen of the hand-held computer, as well as being returned as a character string to the calling program.

The scan times out: The scan beam on time expires and the scanner beam is turned off before the bar code has been decoded. The scanner trigger must be released before another scan can be initiated. This condition can be caused by the following:

- Scanning an unsupported bar code symbology.
- Scanning a disabled bar code symbology.
- Scanning a damaged or otherwise unreadable (out of specification) bar code.
- Scanning something that is not a bar code.

The scan is cancelled: The scanner trigger is released before the bar code has been decoded.

The scan fails: No bar code data is generated by the scan. The cause is not known.

14.5 Scanner Events

The following types of scan events are generated (check the API library documentation for the name of the event and the values returned in each of the development languages):

Event Type	Generated When...	Returns
Scan complete	Scan is successful	
Scan failed	Scan is not successful	Scan cancelled Scan timed-out Scan failed

SERIAL PORTS 15

15.1 Overview 155
15.2 WORKABOUT PRO G2 Serial Port Assignments. 155
15.3 7530, 7535, 8525, and 8530 Serial Port Assignments 156
15.4 8515 Serial Port Assignments. 156
15.5 Ikôn Serial Port Assignments 157
15.6 NEO Serial Port Assignments. 157
15.7 Events 157
15.8 Java 158
15.9 Serial Port API Elements 158

15.1 Overview

The Mobile Devices SDK enables the detection of serial ports on Psion Teklogix computers. It also provides information on these serial ports, and listeners for physical change events that arise on these ports. For .NET, it also provides the ability to read and write data through the serial ports.

Serial ports can be dynamically added to, and removed from, Psion Teklogix computers. Also, on some computers, serial ports can change their physical location. COM ports may be associated with actual physical serial ports, or they may be assigned to a device that acts like a serial port, such as an IrDA port, a USB port, or a *Bluetooth* device.

Serial ports can appear on (attach), or disappear from (detach), a Psion Teklogix computer dynamically. For example, a new serial port appears when a 753x computer is placed into a charger with a port replicator attachment, or if a modem card is inserted into a card slot. These ports can then disappear if the Psion Teklogix computer is removed from the charger, or if the modem card is removed from the card slot. Events are generated when serial ports attach or detach.

BSP ports 1-9 can be used to add a *Bluetooth* virtual COM port. For setup instructions see your Psion Teklogix computer's User Manual.

15.2 WORKABOUT PRO G2 Serial Port Assignments

Table 15.1 Default WORKABOUT PRO G2 Serial Port Assignment

Serial Port	Default Assignment
BSP 1-9	Bluetooth virtual devices
COM0	
COM1	On 100-pin expansion connector.
COM2	Virtual serial port of the tether port. No adaptor is required to use these lines.
COM3	**Cannot be reassigned.** Internal scanner or imager.
COM4	**Cannot be reassigned.** USB client port—used by ActiveSync.
COM5	RS-232 port A on port replicator, and serial port available on USB-to-serial adaptor. This port is removed in suspend and restored on resume.
COM6	RS-232 port B on port replicator. This port is removed in suspend and restored on resume.
COM7	Bluetooth Command Interpreter (blocked) RS-232 port C on port replicator. This port is removed in suspend and restored on resume.
COM8	Virtual port—for WWAN GSM
COM9	**Cannot be reassigned.** IRCOMM port.
COM20	**Cannot be reassigned.** Internal *Bluetooth* radio.
COM21	Built-in USB-Serial adaptor port. This port is removed in suspend and restored on resume.

All COM ports can be reassigned unless indicated otherwise in Table 15.1. Reassignment is done either using the Psion Teklogix COM Port Manager Control Panel program, or in the Windows registry.

For information on using the COM Port Manager, refer to the user manual for the Psion Teklogix computer.

Serial ports on computers with user-accessible cards are assigned dynamically—at the lowest available COM port number—as the cards are inserted and removed.

On the WORKABOUT PRO G2, the maximum baud rate is 921.6 k baud if RTS/CTS hardware flow control is enabled.

Psion Teklogix Serial Endcaps

Psion Teklogix supplies endcaps with serial ports for the WORKABOUT PRO G2 computer. These endcaps use the following serial ports:

- IrDA, TTL, and RS-232 serial endcap (BR1000)—COM9:, COM0:, COM1:, and COM8:
- RS-232 serial endcap (BR1001)—COM9: and COM1:
- IrDA serial endcap (BR1002)—COM9: and COM0:

15.3 7530, 7535, 8525, and 8530 Serial Port Assignments

Table 15.2 Default 753x / 8525 / 8530 Serial Port Assignment

Serial Port	Default Assignment	
	7530 and 7535	8525 and 8530
BSP 1-9	Bluetooth virtual devices	
COM0		
COM1	Tether port—adaptor is not needed.	Tether port—adaptor is not needed.
COM2	Internal scanner or imager. This port is not visible to the SDK.	Serial port.
COM3	Console port on portable docking module.	Console port on the service cable.
COM4	USB client port—used by ActiveSync.	USB client port—used by ActiveSync.
COM5	RS-232 port A on port replicator.	
COM6	RS-232 port B on port replicator.	
COM7	RS-232 port C on port replicator.	

No ports can be reassigned on these computers.

Serial ports on computers with user-accessible cards are assigned dynamically—at the lowest available COM port number—as the cards are inserted and removed.

15.4 8515 Serial Port Assignments

COM1: is the only serial port on the 8515.

15.5 Ikôn Serial Port Assignments

Table 15.3 Default Ikôn Serial Port Assignment

Serial Port	Default Assignment
BSP 1-9	Bluetooth virtual devices
COM0	UMTS
COM2	GPS
COM3	Console port on portable docking module.
COM4	USB client port—used by ActiveSync.
COM5	RS-232 port on Ikôn, and serial port available on the USB port of the port replicator.
COM6	RS-232 port A on port replicator.
COM7	Bluetooth Command Interpreter (blocked).
COM9	UMTS and GSM
COM22	Internal *Bluetooth* radio.
COM23	Internal scanner. This port is not visible to the SDK.

No ports can be reassigned on this computer.

15.6 NEO Serial Port Assignments

Table 15.4 Default NEO Serial Port Assignment

Serial Port	Default Assignment
COM3	Console port (RX and TX data only)
COM4	USB client port—used by ActiveSync.
COM5	USB Serial Port Replicator
COM6	Port Replicator
COM7 / BSP1-9	Port Replicator *Bluetooth* virtual devices
COM21	Internal scanner port
COM22	Internal *Bluetooth* radio.

No ports can be reassigned on this computer.

15.7 Events

The following serial port events are detected by the Mobile Devices SDK:

- Serial port attached events.
- Serial port detached events.
- Serial port location changed events.

Serial port events are only detected while a listener for the specific event type is registered.

Serial port attached event: This is generated when a new serial port is detected on the Psion Teklogix computer.

Examples of situations where this event is generated are as follows: The computer is placed in a cradle or battery charger which has a port replicator; a card—such as a modem card—including a serial port is inserted in a card slot.

Serial port detached event: This is generated when a serial port is removed from the Psion Teklogix computer.

Examples of situations where this event is generated are as follows: The computer is removed from a cradle or battery charger which has a port replicator; a card—such as a modem card—including a serial port is removed from a card slot.

Serial port location changed event: This is generated when the physical location of the serial port associated with a COM number changes on the Psion Teklogix computer.

15.8 Java

Input and output through the serial ports requires the use of one of the following:

- A third party serial port SDK, such as the SerialPort product from Serialio.com.
- The JNI (Java Native Interface) serial classes.

15.9 Serial Port API Elements

C++: Serial port information on all Psion Teklogix Windows computers is obtained using the **PsionTeklogix::SerialPorts** namespace. Input and output uses the standard Win32 API serial communications subset that is available for Windows.

Java: Serial port information on all Psion Teklogix Windows computers is obtained using the **com.teklogix.serial** package.

.NET: Serial port information on all Psion Teklogix Windows computers is obtained, and serial input/output is enabled, using the **PsionTeklogix.Serial** namespace.

SOUND 16

16.1 Sound Hardware 161
16.2 Playing Beeps 161
16.3 Playing WAV Audio Format Files 161
16.4 Sound API Elements 162

16.1 Sound Hardware

The Mobile Devices SDK controls beepers and speakers on Psion Teklogix computers as well as playing beeps and waveform audio. Beepers, which are capable of providing louder sounds than speakers, are useful in environments with high background noise levels. Beepers and speakers are available as follows:

Psion Teklogix Computer	Beeper	Speaker
7530[1]	Yes	Yes
7535	Yes	No
8515	Yes	No
8525	Yes	No
8530	Ye	No
Ikôn	No	Yes
NEO	Yes	No
WORKABOUT PRO	No	Yes

[1]The 7530 has both a beeper and a speaker. When an application plays a beep on a 7530, it is played on the beeper and no attempt is made to play it on the speaker.

16.2 Playing Beeps

The following table defines the valid ranges for parameters that define beeps in the Mobile Devices SDK:

Parameter	Range
Frequency Range (Hz)	800 to 3000
Beep Duration (ms)	1 to 2000
Volume (% of maximum)[1]	0 to 100

[1]On devices with beepers only, this value is overridden by the volume setting in the Control Panel and has no effect. The volume value specified in the API call is only used on devices with waveform audio support through a speaker.

16.3 Playing WAV Audio Format Files

The Mobile Devices SDK enables the playing of WAV audio format files on Psion Teklogix computers that have a speaker. Any valid WAV file that fits into memory can be played.
Psion Teklogix speakers can play all audible frequencies.

WAV files are played by specifying one of the following:

- A WAV file.
- A WAV file contained in a resource file—a JAR file or a ZIP file.
- A system sound.
- Waveform audio data contained in memory.

The following options are available when a WAV file is played:

- Asynchronous—the sound is played asynchronously and the call returns immediately after beginning the sound. To terminate an asynchronously played waveform sound before it is complete, either another waveform sound must be started, or a call must be made to an API element that stops the playing of sounds.
- Synchronous—the sound is played synchronously and the call returns when the waveform sound is complete.
- Loop—the call returns immediately after beginning the sound and the sound is played repeatedly. To terminate a looping waveform sound either another waveform sound must be started or a call must be made to an API element that stops the playing of sounds.

Warning: If an application terminates without stopping a looping waveform sound, the sound will continue to play. The sound is terminated when another application calls the API element that stops the playing of sounds.

The WAV file can be located anywhere in the file system. When specifying a filename as a parameter, the full filename and path should be included in the string. If the path is not specified, the Windows default pathnames are searched.

A WAV file can be played from a memory card inserted in one of the card slots on the Psion Teklogix computer. In this case, the folder name representing the card must be included in the filename string. The filename string, in this case, will be similar to one of the following:

- sd-mmc\wavefile.wav
- hard disk\wavefile.wav
- storage card\wavefile.wav

16.4 Sound API Elements

C++: The beeper on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix::Sound** namespace. Use the Microsoft® Win32 APIs to control the playing of WAV files.

Java: The playing of WAV files and the beeper on all Psion Teklogix Windows CE computers is controlled using the **com.teklogix.sound** package.

.NET: The playing of WAV files on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix.Sound** namespace. The beeper on all Psion Teklogix computers is controlled using the **PsionTeklogix.Sound.Beeper** namespace.

SYSTEM INFORMATION 17

17.1 System Information 165
17.2 Machine Type 165
17.3 Model 166
17.4 Unique Machine Identifier 166
17.5 Psion Teklogix Build Codes. 166
17.6 Setting The Ratio Of Program Memory To Storage Memory 166
17.7 System Information API Elements 167

17.1 System Information

The Mobile Devices SDK provides Psion Teklogix-specific hardware and configuration information.

The following are available if the information is not accessible using Windows calls from the development language:

- The Operating System version and build code.
- The current language.
- Psion Teklogix computer type.
- Psion Teklogix model number.
- Unique Machine Identifier (UID).
- The processor name.
- The processor speed.
- Total amount of flash memory.
- Amount of free flash memory available for use by applications.
- Amount of flash memory in use by applications and the operating system.
- Total amount of RAM.
- Amount of free RAM available for use by applications.
- Amount of free memory available for data storage.
- Build code of the Windows CE image.
- Build code of the peripheral controller software (PCON).
- Build code of the boot (BooST) code.
- The device name configured in System Properties on the Control Panel.
- The device description configured in System Properties on the Control Panel.

The following items are available for Java only:

- The name configured in Owner Properties on the Control Panel.
- The company configured in Owner Properties on the Control Panel.
- The address configured in Owner Properties on the Control Panel.
- The home phone configured in Owner Properties on the Control Panel.
- The work phone configured in Owner Properties on the Control Panel.

17.2 Machine Type

The machine type is one of the following strings:

- **Psion Teklogix 7505** (Ikôn)
- **Psion Teklogix 7527C** (WORKABOUT PRO G2-C)
- **Psion Teklogix 7527S** (WORKABOUT PRO G2-S)
- **Psion Teklogix 7530**
- **Psion Teklogix 7535**
- **Psion Teklogix 8515**
- **Psion Teklogix 8525**
- **Psion Teklogix 8530**
- **Psion Teklogix PX750** (**NEO**)

17.3 Model

The Psion Teklogix model is one of the following strings:

- **7505** (Ikôn)
- **7527C** (WORKABOUT PRO G2-C)
- **7527S** (WORKABOUT PRO G2-S)
- **7530**
- **7535**
- **8515**
- **8525**
- **8530**
- **PX750** (**NEO**)

17.4 Unique Machine Identifier

The Unique Machine Identifier (UID) is a null-terminated string. The string has the format:

- <hardware serial number>-<terminal serial number>

The hardware serial number is supplied by the board manufacturer. The terminal serial number is entered during manufacture. Both are 12 bytes in length and are returned to the caller as a **null** terminated string with a '-' separating the serial numbers.

The 753x computers have both serial number strings truncated to 10 digits to maintain compatibility with earlier releases.

17.5 Psion Teklogix Build Codes

Several system information calls return Psion Teklogix software versions —also called build codes or date codes. These codes are constructed as follows:

1. [A-L]—a single letter representing the month of the year from January to December.
2. [01-31]—a two-digit number representing the day of the month.
3. [0-9]—a single digit representing the year.
4. [a-x]—a single letter representing the hour when the software was built.

e.g. B058n = February 05, 2008, 2:00 pm

Note: This date code refers only to the specific time that the installed software was compiled. More recent date codes do not imply more advanced or more stable software builds. This code should be used only for purposes of identifying the specific software builds and comparison against other builds on a simple equality/inequality basis.

17.6 Setting The Ratio Of Program Memory To Storage Memory

The Mobile Devices SDK enables the setting of the ratio of the amount of memory used for running programs to the amount of memory used for the object store. A low ratio means more memory is dedicated to programs. A high ratio reserves more memory for the object store, and less for programs. For example, setting this ratio to 40% allocates 40% of unused memory to data storage and the remaining 60% to program storage.

Both memory amounts must be at least 256 kilobytes.

17.7 System Information API Elements

C++: The system information on all Psion Teklogix Windows CE computers is accessed using the **PsionTeklogix::System::SystemInformation** namespace.

Java: The system information on all Psion Teklogix Windows CE computers is accessed using the **SystemInformation** class in the **com.teklogix.system** package.

.NET: The system information on all Psion Teklogix Windows CE computers is accessed using the **SystemInformation** class in the **PsionTeklogix.SystemPTX** namespace.

TOTAL RECALL 18

18.1 Total Recall 171
18.2 Cold Resets And Clean Starts 171
18.3 Backup Profile 171
18.4 Profile Types 171
18.5 Profile Content 171
18.6 Storage Locations For Profiles 172
18.7 Restoring Profiles 172
18.8 Troubleshooting 173
18.9 Total Recall API Elements 173

18.1 Total Recall

Total Recall is a Psion Teklogix utility for backing up and restoring files and configurations. Its primary use is to save a copy of data which would otherwise be lost during a cold reset of the device to a permanent or removable storage medium. This allows the data to be restored when necessary.

Previously, Total Recall was also used for purposes of cloning data from one device to another similar device. This functionality has been deprecated.

Total Recall can be run either using the Mobile Devices SDK, using a Control Panel program, or from the command line.

At this time, using the Mobile Devices SDK the only form of backup and restore that is available is a complete backup of all files, databases, and the registry to the default folder. Selection of components to back up is not available. Profiles can only be autorestored from the default folder.

18.2 Cold Resets And Clean Starts

A clean start should only be performed by Psion Teklogix personnel. This returns the registry to its factory settings and removes all files from flash. For more details on reset and clean start procedures and effects, see Section 1.5: “Resetting Psion Teklogix Computers”.

18.3 Backup Profile

A Total Recall backup profile is a collection of files and/or registry settings that can be manually or automatically restored. A complete description of Total Recall for each Psion Teklogix computer is included in the user manual for each computer. This section provides an overview of Total Recall.

18.4 Profile Types

The are three types of Total Recall profile:

- For this device only — a backup that is manually restored by the operator.
- AutoRestore for this device only — a backup that automatically restores itself after a cold reset or a clean start.
- AutoRestore for this and other devices **(deprecated)** — a backup that restores itself after a cold reset or a clean start. This does not include the touchscreen calibration or the wireless radio settings. This profile can be used to clone files and settings from one computer to another.

18.5 Profile Content

A profile contains one or more of the following:

- Files
- Databases
- Registry settings

Files

This option allows all the files on the device (excluding system files) or only a selected list of files to be included in the backup.

Databases

When databases are selected for backup, all eligible databases on the Psion Teklogix computer are backed up. Only Windows-based databases are backed up when this option is selected: third-party databases are not backed up.

If all files are selected for backup (see above), then the database files are included in the backup.

Registry Settings

When this option is selected, all the registry settings are backed up.

18.6 Storage Locations For Profiles

Profiles can be stored in any external or permanent storage medium, such as:

- Flash Disk
- SD card

The location and the name of the profile are specified when the profile is created. However, auto-restore only restores profiles from the default directory. The name of the profile is specified when it is created. When the profile is created by the Mobile Devices SDK, if another profile with the selected name already exists, Total Recall will assign a default name to the profile.

Default Storage Locations

The default storage location for the profile depends on the Windows version and the available hardware:

Windows CE:	\Flash Disk
Windows Mobile:	\SD Card if available, otherwise \My Documents\Total Recall

The default profile filename is *Default1.pfl*. If this file already exists when you try to create a new profile, the numeric component will be automatically incremented until an available filename is found (i.e. *Default2.pfl*, *Default3.pfl*, etc.).

18.7 Restoring Profiles

The following methods are available for restoring saved profiles:

- Manually through the Control Panel
- Programmatically using the Mobile Devices SDK
- Automatically following a cold reset or a clean start
- From the command line—not available in Windows Mobile

The restoration method is selected when the profile is created; however, autorestore profiles may be manually or programmatically restored.

When an autorestore occurs, the active autorestore profile is restored. Normally, this is the most recently created autorestore profile. The Mobile Devices SDK allows the active autorestore profile to be selected from the autorestore profiles stored in the default directory. When an autorestore profile is selected to be the active autorestore profile, its timestamp is updated to make it appear to be the most recent profile.

18.8 Troubleshooting

When Total Recall performs backup and restore operations, the progress and results are written to a log file named *log.txt*, located in the root directory of the computer. If any errors occur in the operation, this file will contain information helpful in determining the nature of the error and the point in the process where the error occurred. This file may be requested by Psion Teklogix support personnel to assist in troubleshooting problems with Total Recall operations.

18.9 Total Recall API Elements

C++: Total Recall on all Psion Teklogix Windows computers is controlled using the **PsionTeklogix::TotalRecall** namespace.

Java: Total Recall on all Psion Teklogix Windows computers is controlled using the **com.teklogix.totalrecall** package.

.NET: Total Recall on all Psion Teklogix Windows computers is controlled using the **PsionTeklogix.TotalRecall** namespace.

WINDOWS SHELL 19

19.1 Windows Shell 177
19.2 Setting Windows Security. 177
19.3 Enabling And Disabling The Windows Shell. 177
19.4 Security Level Change Event 177

19.1 Windows Shell

On Psion Teklogix computers, the user's level of access to the Windows shell can be restricted to prevent access to unnecessary or potentially harmful aspects of the system.

The following Windows shell features can be set and queried by the Mobile Devices SDK:

- Setting Windows security
- Enabling and disabling shell access

19.2 Setting Windows Security

Psion Teklogix computers have the following Windows shell security levels:

- User
- Supervisor
- Teklogix

These levels can also be controlled through the Security item on the Windows **Start** menu.

User security level: This is the most restrictive security level. When this level is set on a Psion Teklogix computer, the computer can be used to run applications, but there is limited access to terminal configuration features. No password is needed to access a computer when it is in this mode.

In user mode, the following restrictions apply:

- On the Windows **Start** menu, the Programs, Settings, and Run items are not available.
- If the Windows desktop is accessible, the **My Computer** and **Recycle Bin** icons are not available.

Supervisor security level: This security level allows access to most of the configuration features of the Psion Teklogix computer. The supervisor password can be changed when the computer is in supervisor mode or in Teklogix mode.

Teklogix security level: This security level gives unrestricted access to the Psion Teklogix computer. The Teklogix password cannot be changed.

19.3 Enabling And Disabling The Windows Shell

The Mobile Devices SDK provides applications with the ability to enable and disable the Windows shell. By default, the shell is enabled. When the shell is disabled, the following features are **not** available:

- The Windows **Start** button
- The Windows Icon Tray
- The Windows Task Manager

The shell must be enabled before the application exits. If the application terminates leaving the shell disabled, the shell can only be enabled again by a warm reset or a cold reset.

19.4 Security Level Change Event

An event is generated when the Windows security level is changed through the Mobile Devices SDK or by the user through the configuration dialog. This event is only detected while a listener for the event is registered.

19.5 System Security API Elements

C++: The system security on all Psion Teklogix computers is controlled using the **PsionTeklogix::System::Security** namespace.

Java: The system security on all Psion Teklogix computers is controlled using the **Security** class and the **ShellSecurityLevelChangeEvent** class in the **com.teklogix.system** package.

.NET: The system security on all Psion Teklogix computers is controlled using the **Security** class in the **PsionTeklogix.SystemPTX** namespace.

TRIGGER CONTROL 20

20.1 Overview 181
20.2 Definition Of Terms 181
20.3 Trigger Consumer Registration 181
20.4 Trigger Source IDs 181
20.4.1 Virtual Key Codes 182
20.5 Trigger Associations 182
20.5.1 Trigger Control Flags 182
20.6 Double-Clicks 183
20.7 Events 183
20.7.1 Simulated Events 184
20.7.2 Single-Click Events And Double-Click Events 184
20.8 Trigger Control API Elements 184

20.1 Overview

All Psion Teklogix computers have one or more keyboard [SCAN] buttons, some have [SCAN] buttons on the side, and some have the option for a pistol grip trigger. Usually, these buttons are all used to initiate a bar code scan or an RFID read cycle. The Mobile Devices SDK allows you to select any keyboard key or the pistol grip trigger as the means of controlling hardware activity. Ownership of the trigger mechanisms can be defined in the Mobile Devices SDK.

Each time a trigger is pressed or released an event is generated. The Mobile Devices SDK also enables an application to generate simulated trigger events.

20.2 Definition Of Terms

Trigger source: A trigger source causes a trigger event. A trigger source is one of the following:

- The pistol grip trigger.
- Any key on the keyboard of the Psion Teklogix computer, including any [SCAN] buttons.

Trigger source ID: Each trigger source has a unique trigger source ID in the form of an integer.

Friendly name: Each trigger source can have a user-friendly name. The friendly name appears on the Manage Triggers Control Panel applet, and only trigger sources with a friendly name defined can be accessed through the applet. This name—maximum 15 characters— is the name of the keyboard key or the trigger source, for example:

- Grip Trigger
- F1
- Scan
- L. Side Scan

If your application creates a new trigger source, and you want to access the trigger source through the Manage Triggers applet, you must give it a friendly name.

Trigger consumer: An application that receives trigger events. Each trigger consumer is identified by its unique name—a character string.

Trigger association: A mapping between a trigger source and a trigger consumer.

Double-click time: This is the maximum length of time allowed between two successive presses of a trigger source for a double-click event to be generated. The default value is 0 (zero).

20.3 Trigger Consumer Registration

Each application that is a trigger consumer must be registered to receive the trigger events. During registration, a unique trigger consumer name must be associated with the trigger consumer. Once it is registered, it can be associated with one or more trigger sources. Deregistering a trigger consumer makes all its trigger associations inactive—they become active again when the consumer registers again.

20.4 Trigger Source IDs

A trigger source ID is associated with the pistol grip trigger or one of the keyboard keys using keyboard remapping. See "Keyboard Remapping" on page 56 for information on how to do this.

Trigger source IDs are unique. The first 256 ID numbers (0-255) are reserved as system (non-keyboard) trigger IDs. Most system trigger IDs will not be available to applications (e.g. most external scanner triggers). The most notable exception is the pistol-grip trigger, which has the system trigger ID value of 0 (zero).

Keyboard trigger sources start at 256—0x100. The trigger source ID for a specific key is 256 plus the virtual key code. See "Keyboard Remapping" on page 56 for information on virtual key codes. For example the virtual key code for the [F1] key is 40. So the trigger source ID for the [F1] key is 296; the sum of 256 and 40.

20.4.1 Virtual Key Codes

See the following for a list of virtual key codes:

C++: PsionTeklogix::Keyboard::Key enumeration

Java: com.teklogix.keyboard.VirtualKey class

.NET: PsionTeklogix.Keyboard.Key enumeration

20.5 Trigger Associations

In order to receive trigger events, a trigger consumer must be associated with one or more trigger sources. Some hardware components should not be operated simultaneously, due to their heavy power consumption. For example, an internal imager and a tethered RFID reader should not be scanning at the same time. Doing so may cause the terminal to spontaneously turn off due to insufficient power.

Before being associated with the trigger source, the trigger consumer must register to receive trigger events. Only the trigger consumers associated with a specific trigger source receive events originating with that trigger. Several consumers can be associated with the same trigger source.

When a trigger consumer is deregistered, all its trigger associations are deactivated. If the trigger consumer re-registers the associations are re-activated.

Each development language provides API elements that associate a trigger source ID with a trigger consumer.

20.5.1 Trigger Control Flags

Trigger control flags modify a trigger association. These flags are specified when a trigger association is created. Several trigger control flags can be set during one trigger association attempt.

Unless the temporary flag is set, all trigger associations are permanent and persist across both warm resets and cold resets. It is recommended that unless the application is launched at system startup, the temporary flag be set for all trigger associations.

Exclusive flag: Setting this flag ensures that only one trigger consumer is associated with a trigger source. If this flag is set during an attempt to create a trigger association, the following can occur:

- If there is no existing trigger association for the trigger source, the association is successful and all further attempts to create an association for this trigger source fail until this exclusive association is deleted.
- If there is an existing trigger association for the trigger source, this attempt to create an exclusive association fails.

Override flag: When this flag is set during an attempt to create a trigger association, all existing trigger associations for the trigger source are replaced by the new one. Even an exclusive trigger association is replaced.

Temporary flag: When this flag is set, the trigger association does not persist across either a warm reset or a cold reset. If the override flag is also set, then the previous trigger association for the trigger source is restored when this association is deleted, or the trigger consumer is deregistered. Only one temporary-plus-override flag can be in existence for each trigger source.

Ignore duplicate registration flag: When this flag is set during a trigger association attempt, if the trigger association already exists, no error is returned.

Wants-trigger-events flag: When this flag is set, the consumer receives trigger-down events and trigger-up events. It does not receive double-click events unless the wants-double-click-events flag is also set, or there are no consumers registered to receive double-click events.

Wants-double-click-events flag: When this flag is set, the consumer receives double-click events. To receive single-click events too, the wants-trigger-events flag must also be set.

20.6 Double-Clicks

A double-click occurs when the pistol grip trigger or another trigger source is pressed twice within a very short time. This time-gap is measured in milliseconds. The double-click time is the maximum time allowed between the trigger presses if a double-click event is to be generated. The default double-click time is 0 (zero).

A double-click event is generated when both the following are true:

- A trigger source is pressed and released within ½ the configured double-click time, and
- The trigger source is pressed a second time before the double-click time expires.

Each trigger source has its own double-click time.

The trigger that is associated with the internal scanner uses the double-click time configured for the scanner either through the Control Panel or through the SDK scanner namespace. See "Double-click Settings" on page 90 for more information.

20.7 Events

To receive trigger events, a trigger consumer must be associated with a trigger source and registered to receive trigger events. A trigger consumer can receive events from several triggers. Each event contains the identity of the trigger source that originated it. To receive trigger events an application must complete the following steps:

1. Register as a trigger consumer.
2. Register the trigger consumer to receive trigger events.
3. Associate the trigger consumer with a trigger source ID.

A trigger consumer can deregister as a trigger consumer to stop receiving all events. It can remove the association with one, or all, trigger source IDs to stop receiving events from one, or all, trigger sources.

A trigger event is generated when the state of a trigger source changes on a Psion Teklogix computer. There are two trigger states:

- Trigger-down.
- Trigger-up.

Trigger-down event: This event is generated when the pistol grip trigger or the button associated with a trigger source is pressed.

Trigger-up event: This event is generated when the trigger or button associated with a trigger source is released.

Any trigger event can have one of the following flags set:

- Single-click flag.
- Double-click flag.

Single-click flag: This flag is set when a trigger source is pressed and released within ½ the double-click time configured for that trigger source. The double-click time must be greater than 0 (zero).

Double-click flag: This flag is set when a trigger source is pressed twice within the double-click time configured for that trigger source. The double-click time must be greater than 0 (zero).

20.7.1 Simulated Events

The Mobile Devices SDK can generate a simulated trigger event. This is forwarded to the trigger driver. The trigger driver sends the event to all registered trigger consumers which also have a trigger association with the specified trigger event source.

This method generates both trigger-up and trigger-down events.

Any application that simulates trigger events using this method **must** be well behaved. A trigger-down event must always be followed by a trigger-up event. Sending out multiple trigger-down and trigger-up events in a row can result in unexpected behaviour. An application must not remove a trigger association after a trigger-down event—the associated trigger-up event must be simulated before the association is ended.

20.7.2 Single-Click Events And Double-Click Events

Single-click events and double-click events can be generated by any trigger source, if the configured double-click time for the trigger source is not 0 (zero).

Event Sequence (> = followed by)	Generated When...
Trigger-down event > trigger-up event with the single-click event flag set.	A trigger source is pressed and released within ½ the configured double-click time.
Trigger-down event > trigger-up event with the single-click event flag set > trigger-down event with the double-click flag set > trigger-up event with the double-click flag set.	A trigger source is pressed and released within ½ the configured double-click time, and it is pressed and released a second time before the double-click time expires.
Trigger-down event > trigger-up event with the single-click event flag set > trigger-down event with the double-click flag set > trigger-up event.	A trigger source is pressed and released within ½ the configured double-click time, and it is pressed a second time—but not released—before the double-click time expires.

20.8 Trigger Control API Elements

C++: The trigger on all Psion Teklogix computers is controlled using the **PsionTeklogix::Trigger** namespace.

Java: The trigger on all Psion Teklogix computers is controlled using the **com.teklogix.trigger** package.

.NET: The trigger on all Psion Teklogix computers is controlled using the **PsionTeklogix.Trigger** namespace.

WIRELESS LOCAL-AREA NETWORKING 21

21.1 Wireless Local-Area Networking (WLAN) 187
21.1.1 Radios Supported 187
21.1.2 Configuring WLAN Through The Control Panel 187
21.1.3 Wireless Zero Config 187
21.1.4 Summit Client Utility 187
21.2 Availability Of API Libraries 188
21.3 Encryption For Data Transmission 188
21.3.1 Wired Equivalent Privacy (WEP) Keys. 188
21.4 Connection State Parameters 188
21.4.1 Received Signal Strength Indicator—RSSI. 188
21.4.2 Transmit Power 188
21.4.3 Network adaptor Name 189
21.5 Summit Radio Features 189
21.5.1 Summit Mode 189
21.5.2 Wireless Zero Config Mode (WZC) 189
21.5.3 Changing Between Modes 189
21.6 WLAN API Elements. 189

21.1 Wireless Local-Area Networking (WLAN)

A wireless LAN enables a device to connect to a local-area network (LAN) through a wireless radio connection. The IEEE 802.11 standards define WLANs. The coverage area of a WLAN is dependant on the radio wave frequency, power output and environmental factors that may reflect or absorb radio signals.

The Mobile Devices SDK provides the following features:

- A straightforward way for a software application to connect a Psion Teklogix computer with an access point.
- A generic interface that enables a radio connection to be made without a detailed knowledge of the specifics of the radio.

21.1.1 Radios Supported

The Mobile Devices SDK WLAN is supported by all the Psion Teklogix 802.11 radios:

21.1.2 Configuring WLAN Through The Control Panel

The following utilities can be used to configure WLAN connections on Psion Teklogix computers:

- Wireless Zero Config (WZC)
- Summit Client Utility (SCU)
- Odyssey Access Client by Juniper Networks (OAC)

21.1.3 Wireless Zero Config

Wireless Zero Configuration (WZC) is the Windows service that automatically configures 802.11 wireless network devices.

WZC is a standardized set of Microsoft interfaces for wireless network cards. If the driver interfaces with WZC, it can be controlled and queried through this standardized interface, making configuration and status querying consistent—regardless of the manufacturer of the adaptor. Not all wireless cards are WZC–compatible. The WZC interface is not well documented.

Microsoft provides an API library for Wireless Zero Config.

21.1.4 Summit Client Utility

The Summit Client Utility (SCU) is an application for end users and administrators of mobile devices that use a radio manufactured by Summit Data Communications.

The SCU provides a graphical user interface (GUI) for access to all its functions. Access to these functions is also available through an API library. Using the APIs, an application such as Wavelink Avalanche can manage Summit radios.

The Summit 802.11 radio *User's Guide s for Windows CE.NET and Windows Mobile* is available at http://www.summitdatacom.com/documentation.htm

21.1.5 Odyssey Access Client by Juniper Networks

The Odyssey Access Client by Juniper Networks (formerly Funk Software) is an application that can be used in conjunction with a RADIUS server to establish secure authentication for network connections.

There are currently no APIs for managing the Odyssey Access Client through an application.

21.2 Availability Of API Libraries

The WLAN API libraries are available as follows:

API Library	APIs Available On...		
	Summit Radio	RA2040	RA2043
Mobile Devices SDK WLAN APIs	Yes	Yes	Yes
Summit Client Utility	Yes	No	No
Wireless Zero Config	Yes	Yes	Yes
Odyssey Access Client	No	No	No

21.3 Encryption For Data Transmission

21.3.1 Wired Equivalent Privacy (WEP) Keys

Wired Equivalent Privacy (WEP) is an encryption algorithm system and part of the 802.11 standard. It is a security measure to protect wireless LANs from casual eavesdropping. WEP uses a shared secret key to encrypt packets before transmission between wireless LAN devices, and it monitors packets in transit to detect attempts at modification.

WEP Key Length

The length of the WEP key determines the size of the encryption key:

- A 5-character text string, or a 10-character hexadecimal string, gives a 40-bit encryption key.
- A 10-character text string, or a 26-character hexadecimal string, gives a 128-bit encryption key.

WEP Key Index

The Mobile Devices SDK can store up to four WEP keys. The index is used to identify each key (1 to 4) when determining which key to use.

21.4 Connection State Parameters

21.4.1 Received Signal Strength Indicator—RSSI

The RSSI status of the radio changes continuously based on a number of different environmental factors, such as distance, interference, angle of antenna, etc. The RSSI is measured in dBm. A perfect signal is -10 dBm. An complete absence of signal is measured as -200 dBm.

21.4.2 Transmit Power

The Mobile Devices SDK enables the transmit power of the radio to be set and queried. This level is expressed as a percentage of the maximum transmit power of the radio. The available power levels increase incrementally; so not all percentages are valid. If you try to set the level to a non-valid percentage, the value is increased to the next valid percentage. Similarly, when you query the setting, the returned value is rounded up to the next valid percentage.

The maximum transmit power of the radio is regulated by the country in which the radio is to be used.

21.4.3 Network adaptor Name

The network adaptor name can be queried using the Mobile Devices SDK. The string returned is the same as the adaptor name returned by the Control Panel on the same Psion Teklogix computer.

21.5 Summit Radio Features

Radios manufactured by Summit behave differently from the other radios available for Psion Teklogix computers. Summit radios operate in one of the following modes:

- Summit mode.
- Wireless Zero Config mode—referred to as Third Party Config mode.

21.5.1 Summit Mode

When operating in this mode, the Summit radio does not appear on the list of radios controlled by the ***Wireless*** Control Panel applet. Instead it is controlled by the ***Summit*** Control Panel applet.

The preferred list of access points is not available, so the radio cannot roam between networks; however it can roam between access points on the same network.

21.5.2 Wireless Zero Config Mode (WZC)

When operating in this mode, the Summit radios behave in the same way as all the other radios that are available on Psion Teklogix computers.

21.5.3 Changing Between Modes

When the operating mode of a Summit radio is changed between Summit mode and Wireless Zero Config mode, the radio must be reset by powering it down, and then powering it up.

21.6 WLAN API Elements

C++: WLAN on all Psion Teklogix computers is controlled using the **PsionTeklogix::WLAN** namespace.

Java: WLAN on all Psion Teklogix computers is controlled using the **com.teklogix.wlan** package.

.NET: WLAN on all Psion Teklogix computers is controlled using the **PsionTeklogix.WLAN** namespace.

WIRELESS WIDE-AREA NETWORKING 22

22.1 Wireless Wide-Area Networking (WWAN) 193
22.2 Psion Teklogix Computers Supporting WWAN 193
22.3 Supported WWAN Modems 193
22.3.1 Multiplexing. 194
22.3.2 Virtual Serial Port 194
22.3.2.1 Using The Virtual Serial Port 194
22.3.2.2 Configuration 194
22.3.2.3 AT Commands 195
22.3.2.4 Entering AT Commands in Windows CE 195
22.3.3 Dial-up Data Connections. 196
22.3.3.1 Dial-up Data in GSM Networks 196
22.3.3.2 Dial-up Data in UMTS Networks 197
22.3.3.3 Dial-up Data in CDMA Networks 197
22.3.3.4 Dial-up Data in iDEN Networks. 197
22.3.4 Packet Data 197
22.3.5 Summary Of Modem Differences 198
22.3.6 GSM Power Driver 198
22.4 Initializing WWAN 199
22.4.1 Checking The Initialization Status Of The WWAN Driver. 199
22.4.2 WWAN Driver Status Flags. 199
22.4.3 Initializing The WWAN Driver 201
22.5 Closing WWAN 205
22.6 Connecting To The Internet 205
22.6.1 Windows CE 5.0 Connection 206
22.6.2 Windows Mobile 5.0 And Windows Mobile 6 Connection 209
22.6.3 The DbGprs.csv File 209
22.6.4 Checking Packet Data Status 210
22.6.5 Roaming. 210
22.6.6 Connecting Manually to A WWAN Network 210
22.6.7 Selecting An Access Point Name (APN) 211
22.6.8 Connecting To A VPN 211
22.7 Access Flags. 212
22.7.1 Controlling The Interaction With The Control Panel 213
22.8 RSSI (Received Signal Strength Indicator) 213
22.9 WWAN API Elements 214
22.10 Using SMS (Short Message Service) 215
22.10.1 SMS API Elements 215
22.11 WWAN Supplementary Services 216
22.11.1 Voice Service States 217
22.12 Voice Calls On A WWAN 217

22.12.1 Initializing WWAN 217
22.12.2 Initializing Voice Over WWAN 217
22.12.3 Closing Voice Over WWAN 220
22.12.4 Making A Phone Call 220
22.12.5 Voice Call States 222
22.12.6 Receiving A Phone Call 224
22.12.7 Processing Voice Calls 225
22.12.8 Terminating Voice Calls 227
22.12.9 Call Forwarding 227
22.12.10 Dual-Tone Multifrequency (DTMF) 228
22.12.11 Blocking Inbound And Outbound Calls 228
22.12.12 Blocking Caller ID On Outgoing Calls 228
22.12.13 Audio for Voice Over WWAN 228
22.12.14 Voice Over WWAN Events 229
22.12.14.1 Voice Call Status Changed Event 229
22.12.14.2 Voice Call Manager Incoming Call Event 230
22.13 Phone Books 230
22.13.1 Data Coding Schemes For Phone Book Entries 231
22.13.2 Phone Book Types 232
22.13.3 Phone Book API Elements 232
22.14 Resource Materials 233

22.1 Wireless Wide-Area Networking (WWAN)

The Mobile Devices SDK gives access to WWAN features that are available to a computer using a radio-based network, but which are not available to, or are not needed by, a computer on a wired network.

Wireless Wide-Area Networks are wireless networks that cover large geographic areas. Wireless Wide-Area Networks are public cellular networks based on technologies such as GSM, UMTS, CDMA, or iDEN.

WWANs are also referred to as wireless broadband and broadband wireless networks.

The Mobile Devices SDK provides access to WWAN capabilities implemented by Psion Teklogix. It supports voice calls and data calls, as well as SMS (Short Message Service) and also gives access to phone books resident on a SIM card or the modem.

22.2 Psion Teklogix Computers Supporting WWAN

The Mobile Devices SDK WWAN capability is supported on Psion Teklogix computers as follows:

	Windows Embedded CE 5.0	Windows Mobile 6 Classic
7535 G2	Yes	No
Ikôn	Yes	Yes
WORKABOUT PRO G2	Yes	Yes

Note: For computers operating under Windows Mobile 6 Professional, WWAN capability and the APIs to access it are provided by Windows.

22.3 Supported WWAN Modems

Psion Teklogix computers support a number of WWAN modems. They come in either PC or CF card form, or they are built-in and connect to an expansion port inside the computer. The Mobile Devices SDK makes it possible to develop applications without having a detailed knowledge of the specific WWAN modem.

The following table lists the WWAN modems that are officially supported on Psion Teklogix computers.

Computer Model	Modems Supported
7535	Asus AGC-100; Enfora ADT0110
Ikôn	Siemens HC25, HC28, MC75
WORKABOUT PRO	Siemens HC25, HC28, MC75

Other third-party WWAN modems are referenced in this chapter and may be made to work, but support for them will be limited. Consult your local Psion Teklogix support representative before proceeding with development on a modem not listed here.

22.3.1 Multiplexing

Most WWAN modems have only one serial port. Some modems can operate in multiplexing mode.

While a modem without a multiplexing mode has a packet connection, it cannot provide status information to the user interface or to an application. Information such as signal strength cannot be monitored.

The WWAN driver tells a suitable modem to switch to multiplexing mode. This creates several virtual serial ports on the single physical serial port. The PPP connection is on one of the virtual ports. AT modem commands and status data can be sent through another virtual port.

22.3.2 Virtual Serial Port

The Mobile Devices SDK makes a virtual serial port available. The virtual serial port is hard-coded to COM8: on all Psion Teklogix computers. It behaves like any other COM port. As soon as the WWAN modem is initialized, the WWAN driver publishes a COM port interface.

22.3.2.1 Using The Virtual Serial Port

The WWAN driver provides a virtual serial port so that other applications such as Windows CE dial-up networking can access the WWAN modem concurrently with the WWAN user interface. The main purpose is to support dial-up data connections in a more convenient manner. Compared to previous software releases the addition of the virtual serial port offers a number of advantages:

- The WWAN driver does not need to be shut down in order to establish a dial-up data connection
- For some modems where dial-up data connections were not possible before because of peculiarities of the modem they are now possible through the virtual port (Asus AGC-100; Novatel U530, U630).
- For a number of modems the network status display and SMS functions remain available during a dial-up data connection (Siemens MC75, Option Globetrotter, Option Globetrotter Combo, Sierra AC775, Audiovox RTM-8000, CMCS NTN-000, Enfora ADT0110, Enfora GSM0110).
- The WWAN driver and user interface provide the PIN handling.
- For all GSM/GPRS modems the APN for the virtual port can be automatically configured through the WWAN user interface.
- The WWANDbg utility can be used to analyze the AT commands sent by legacy third-party applications.

22.3.2.2 Configuration

The virtual serial port is available as COM8: with the name **Virtual WWAN port**. The port is created dynamically if a WWAN modem is present and the modem initialization was successful (including successful PIN authentication, if required). The virtual port should be configured as follows:

- Any baud rate (this setting is ignored, the correct baud rate for the modem is chosen by the WWAN driver).
- 8 data bits, no parity, 1 stop bit.
- No flow control (flow control for communication with the modem is handled by the WWAN driver).

22.3.2.3 AT Commands

The virtual serial port can be used by an application to send AT commands to the modems. Since AT command sets differ between modems a detailed knowledge of the modem in question is required. Therefore, this use of the virtual port should be limited to legacy applications. New applications should use the Psion Teklogix Mobile Devices SDK to access the network status, SMS functions, etc.

The Option Globetrotter (v1, v2, Combo) and Sierra AC775 do not forward unsolicited AT command responses (e.g. "+CREG:" responses that were enabled with "AT+CREG=1") to the virtual port. For the Enfora ADT0110 and GSM0110, Siemens MC75, Option Globetrotter (v1, v2, Combo), as well as Sierra AC775 it is possible to send AT commands to the modem while a packet data connection through the WWAN driver is active; however, for the Option Globetrotter (v1, v2, Combo) a transition of the emulated DTR signal on the virtual port closes that packet data connection.

When porting legacy applications the virtual serial port together with WWANDbg utility can be used to analyze the AT commands sent by that application. With the debug level set to 6 all AT commands are shown in the log.

22.3.2.4 Entering AT Commands in Windows CE

To enter modem AT commands on a Windows CE device, you first need to create and open a console window, as described in the following steps:

1. Go to **Start > Settings > Network...**
2. Double-click on the **Make New Connection** icon.
3. Enter a name for the new console connection in the text box.
4. Click the **Next** button (leave the connection type as *Dial-Up Connection.*
5. In the **Select a modem** drop-down box, select *Virtual WWAN port.*
6. Click the **Configure** button.
7. Set **Flow Control** to *None*, and leave the other **Connection Preference** settings at their default values.
8. Check the **Use terminal window before dialing** checkbox.
9. Click **OK** to close the **Device Properties** window.
10. Click the **Next** button.
11. Enter an arbitrary value (e.g. *1*) for the **Phone number** text box.
12. Click the **Finish** button to complete the connection configuration.
13. Back on the **Network...** screen, double-click the icon for your newly created connection.
14. Click the **Connect** button. The terminal window will open after a few seconds.
15. Type the AT command *ATE1* followed by **ENTER** to enable command echo, which will allow you to see the subsequent commands you type.

The terminal window is now connected to the modem, allowing you to enter AT commands. When you are finished entering commands, close the terminal window.

22.3.3 Dial-up Data Connections

The following dial-up connections are available through the WWAN driver:

- GSM Networks
- UMTS Networks
- CDMA Networks
- iDEN Networks

22.3.3.1 Dial-up Data in GSM Networks

Not all GSM networks support dial-up data connections to land-line modems (e.g. Rogers in Canada has never supported dial-up data connection, Microcell Fido in Canada discontinued dial-up data connection support a few years ago). For those networks that support dial-up data connections no additional configuration should be required for a connection to an analog modem at 9.6 kbps.

Some networks allow faster dial-up connections at 14.4kbps. The AT command "+CBST=14,0,1" has to be added in the WWAN UI settings of the Wireless Manager Control Panel, under **Properties > Configure… > Call Options > Extra Settings** to enable this mode. The Asus AGC-100 modem does not support 14.4kbps.

The even faster High Speed Circuit Switched Data (HSCSD) mode available on a few networks is not supported by any of the modems in Psion Teklogix products.

For ISDN lines, the situation is more complex. Some ISDN modems may accept analog connections as above, while others require the protocols V.110 or V.120 to be used. These also require an additional AT command.

If an ISDN modem requires the protocols X.75 or HDLC, then no connection is possible. Also, many GSM networks do not support V.120 (e.g. Vodafone D2 in Germany) and only the Audiovox RTM-8000 and CMCS NTN-000 support V.120.

For more details on ISDN connections see D. Živadinovic, M. Winkler, Des Surfers Bastelstunde in c't 7/2001, page 228 (in German).

For more details on GSM AT commands see Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); AT command set for 3G User Equipment (UE), 3GPP TS 27.007 version 5.4.0 Release 5, ETSI, 09/2003.

The following table summarizes the required AT commands.

Land-line modem	Speed	Extra Settings
analog	9.6kbps	
analog	14.4kbps	+CBST=14,0,1
ISDN V.110	9.6kbps	+CBST=71,0,1
ISDN V.110	14.4kbps	+CBST=75,0,1
ISDN V.120	9.6kbps	+CBST=39,0,1
ISDN V.120	14.4kbps	+CBST=43,0,1

22.3.3.2 Dial-up Data in UMTS Networks

In general, UMTS modems should not be used for dial-up data connections.

Only the UMTS networks in Japan allow dial-up data connections. As a consequence, for modems that support both GSM and UMTS (Novatel U530 / U630, Sierra AC850 / AC860) dial-up connections are not possible once the modem has switched to UMTS (which it will do whenever UMTS is available and no dial-up connection is already active). Also, dial-up support is very limited in newer firmware revisions for the Novatel U630—features have been removed compared to older firmware revisions.

22.3.3.3 Dial-up Data in CDMA Networks

No additional configuration is required. The modem should select the highest possible speed (9.6kbps or 14.4kbps) automatically.

22.3.3.4 Dial-up Data in iDEN Networks

The AT command "+WS46=23;+FCLASS=0" has to be added under the Control panel **Properties -> Configure… -> Call Options -> Extra Settings**.

22.3.4 Packet Data

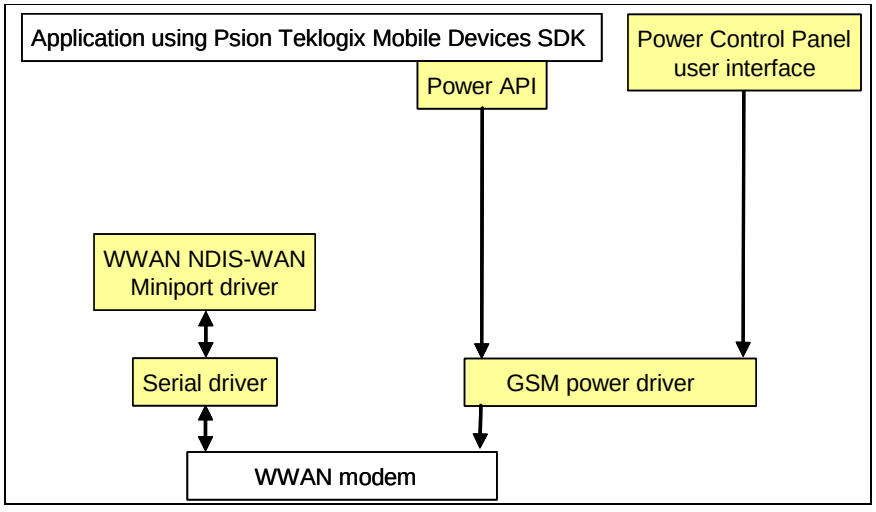
The WWAN driver should be used for packet data connections. A packet data connection through the virtual serial port is slightly less efficient and more difficult to configure. Therefore, only legacy applications that are difficult to change would use the virtual serial port for a packet data connection. The Option Globetrotter (v1, v2, Combo) and Sierra AC775 modems do not allow packet data connections through the virtual serial port. For all GSM modems, the GPRS APN is automatically preconfigured for the virtual port.

22.3.5 Summary Of Modem Differences

Modem Capability	Enfora ADT0110, GSM0110 Siemens MC75	Option Globetrotter v1, v2, Combo Sierra AC775	Audiovox RTM-8000 CMCS NTN-000	Asus AGC-100 GTRAN GPC-6210 Motorola iM240 Novatel U520, U530, U630 Sierra AC550, AC555, AC850, AC860	Siemens HC25, HC28
Dial-up data through virtual port	Yes	Yes	Yes	Yes	Yes
Packet data through virtual port	Yes	No	Yes	Yes	Yes
AT commands through virtual port	Yes	Yes	Yes	Yes	Yes
Unsolicited AT command responses through virtual port	Yes	No	Yes	Yes	Yes
Network status available while virtual port is open	Yes	Yes	Yes	No	Yes
Packet data connection while virtual port is open	Yes	Yes	No	No	Yes

22.3.6 GSM Power Driver

The GSM power driver applies only to built-in radios in the WORKABOUT PRO and Ikôn computers. It is intended that it will work with future expansion board radios. The GSM power driver works in conjunction with the Mobile Devices SDK features and the Control Panel functions that control the power to card slots. Refer to "Card Slots" on page 39 for details.



22.4 Initializing WWAN

Before any of the WWAN functions can be used, you must initialize the WWAN interface.

22.4.1 Checking The Initialization Status Of The WWAN Driver

Use **GetReadyState** to check the initialization status of the driver. This returns one of the following:

- **C++**: A **WWAN_READY_STATE** structure containing the WWAN driver status flags.
- **.NET**: A member of the **WWAN_READY_STATE_FLAGS** enumeration.

22.4.2 WWAN Driver Status Flags

The WWAN status flags indicate the status of the WWAN driver. This WWAN driver status depends on the status of the modem.

Many WWAN API functions depend on the modem being in the ready state. If the modem is not ready, they report an error.

There are two methods for finding the modem status. The first is to use API elements that query the modem status. The second is to register a callback for modem initialization.

This second method—using a callback— is recommended. Repeated queries of modem status can result in excessive processor activity and a consequent drain on the battery power of the Psion Teklogix computer. If there is no other activity on the hand-held computer, the processor can go into an idle state until activated by the callback.

In both cases status bits are returned that indicate which functions are available.

The WWAN driver can be in one or more of the following states:

Shutdown

This flag is set when the WWAN driver detects that the modem has been shut down. If the modem power has been removed through the Control Panel Power applet, or through the SDK then the shutdown flag is expected. In most other cases, the modem requires a repair.

Possible reasons for a modem shutdown are:

- An over voltage has occurred—Siemens HC25 & MC75 only—this should occur only in the case of a hardware failure.
- An over temperature or under temperature has occurred—Siemens HC25 & MC75 only.
- A **SetPinState** has resulted in the modem no longer being accessible; for example, trying to enable/disable/change the PIN too many times with the wrong PIN.

  After rebooting the modem the Personal Unblocking Key (PUK) is required.
- The serial driver has reported an error. One cause of this is that power has been removed from the modem.
- On startup the serial driver could not be opened or the modem never activates the CTS signal.

Phone book ready

This flag must be set for phone book calls to be successful.

SMS store ready

The WWAN driver can send, retrieve, or delete SMS messages. The SMS capabilities or SMS configuration of the modem can be queried. This flag must be set for the following functions to be successful:

- **ReadSms**
- **SendSms**
- **DeleteSms**
- **SetSmsConfiguation**
- **GetSmsConfiguration**

Initialized

The WWAN driver and the modem are ready to access the network.When the driver is in this state the following calls can proceed:

- **SetRegisterState**
- **SetProvisionedContexts**
- **GetAvailableNetworks**

The following functions may return incomplete data, if the driver is not initialized:

- **GetDeviceCaps**
- **GetSIMID**

No SIM

A SIM card is not detected.

Bad SIM

The SIM card is invalid. This is usually a hardware problem. Some modems set this state if the SIM card is missing.

Depending on the startup timing, this status may be returned if the SIM card has not been activated on the network.

If a PIN is entered incorrectly three times in succession, the PUK is requested. If the PUK is entered incorrectly 10 times in succession, the SIM card is permanently disabled, and it must be replaced.

Failure

The modem has failed. This occurs before the modem enters **Shutdown** status. In this state a limited number of query commands are successful. This status usually indicates a hardware failure.

This status is set if the driver cannot communicate with the modem. It is also set if an initialization command, which should always succeed, returns an error.

22.4.3 Initializing The WWAN Driver

*Note: Each **Initialize** call must be paired with a **Shutdown** call. See "Closing WWAN" on page 205 for details.*

Warning: If you call Initialize several times WITHOUT calling the Shutdown method, eventually a call to Initialize will fail. Before WWAN can be initialized again, the Psion Teklogix computer must be reset; a warm reset is recommended.

If **GetReadyState** does not return a WWAN driver status of **Initialized**, the driver must be initialized as follows:

1. Call **Initialize**. During initialization of the WWAN driver and the authentication of the PIN (see PIN Authentication), none of the WWAN driver status flags is set.
2. When the WWAN driver initialization is complete, the **Initialized** flag is set. This occurs even if no network is available.
3. The WWAN driver can now accept any requests except SMS and phone book requests.
4. After a few more seconds, the SMS initialization completes, and the **SMS store ready** flag is set as well as the **Initialized** flag.
5. Phone book initialization can take another minute or more to complete. The **Phone book ready** flag is set alongside any other flags that are already set.

If the SIM card has not been activated on the WWAN network, the **Bad SIM** WWAN driver status error flag may be set, depending on the startup timing.

The SIM card and the modem must wait to be informed by the network of their activation state on the WWAN network. This depends on the network coverage and the timing of messages on the network.

PIN Authentication

To determine if a password is needed, the type of password, and to obtain the password, proceed as follows:

- Call **OnPinState** or **GetPinState** to determine the password type, if the PUK attempts have not been exhausted.

Once the PUK attempts are exhausted, the SIM card can no longer be used. The **Bad SIM** WWAN driver status flag is set.

If there is no SIM card, either the **No SIM**, or the **Bad SIM**, WWAN driver status flag is set.

Problem	WWAN Driver Initialization Status Flag
PUK attempts exhausted.	Bad SIM
No SIM card.	One of the following (depending on the modem type): • No SIM • Bad SIM
The WWAN driver cannot communicate with the modem.	Failure
An initialization command failed.	Failure

Sample Code For Initializing WWAN

This example provides a safe and reliable way to ensure that the WWAN interface is initialized correctly. This example is written in C++; however, the same process can be followed for a .NET application.

Example 22.1 Initializing WWAN in C++

```
// Use this class in the function, InitializeWWANAndWait
class WWANReadyStateListener : public PsionTeklogix::WWAN::IndicationsListener
{
public:
    WWANReadyStateListener()
    {

        // Create an event to wait on, this object signals it when WWAN is ready.
        WwanReadyEvent = CreateEvent(NULL, FALSE, FALSE, NULL);
    }

    ~WWANReadyStateListener()
    {
        if (WwanReadyEvent != 0)
        {
            CloseHandle(WwanReadyEvent);
            WwanReadyEvent = 0;
        }
    }

    HANDLE GetWwanReadyEvent(){ return WwanReadyEvent; }

    virtual void OnReadyState( const WWAN_READY_STATE &ReadyState )
    {
        if ((ReadyState.State & WWAN_READY_STATE_INITIALIZED) != 0 )
        {
            if (WwanReadyEvent != 0)
            {
                SetEvent(WwanReadyEvent);
            }
        }
    }

private:
    HANDLE WwanReadyEvent;
};

bool InitializeWWANAndWait()
{
    WWanRequestParameters params = {0};
    WWAN_READY_STATE* wwanReadyStatePtr = 0;
    WwanRequestStatus retVal;

    // See if the WWAN namespace has been initialized, and get the ready state.
    retVal = WirelessWAN::GetReadyState(params, &wwanReadyStatePtr);
    if(WwanRequestNotInitialized == retVal)
    {

        // The namespace is not initialized, attempt to do so:
        if (WwanResultSuccess != WirelessWAN::Initialize())
        {

            // Error initializing the WWAN namespace.
            return false;
        }
```

Example 22.1 Initializing WWAN in C++ (Continued)

```
        // Try again.
        retVal = WirelessWAN::GetReadyState(params, &wwanReadyStatePtr);
    }

    if ( retVal != WwanRequestSuccess ||
        params.resultCode != WwanResultSuccess ||
        wwanReadyStatePtr == 0)
    {

        // Unable to get the ready state.  See return value/result code for reason.
        return false;
    }

    ULONG readyState = wwanReadyStatePtr->State;

    // Memory was allocated by "GetReadyState(...)"
    free(wwanReadyStatePtr);

    if ((readyState & WWAN_READY_STATE_INITIALIZED) != 0 )
    {

        // Already in the ready state.
        return true;
    }

    // The wwan driver (or modem) is not in a usable state yet.
    // Wait for it to become ready:

    // Create and register the listener object to receive events
    WWANReadyStateListener* myReadyStateListener = new WWANReadyStateListener();
    WirelessWAN::SetIndicationsListener(myReadyStateListener);
    WirelessWAN::SetIndicationsState(true);

    // To prevent a race condition, check to see if the event occured between the
    // time last check for the WWAN ready state and registration for the event.
    retVal = WirelessWAN::GetReadyState(params, &wwanReadyStatePtr); // Try again.
    if( retVal != WwanRequestSuccess ||
        params.resultCode != WwanResultSuccess ||
        wwanReadyStatePtr == 0)
    {

        // Unable to get the ready state.  See return value/result code for reason.
        // Remove indication registration and cleanup objects.
        WirelessWAN::SetIndicationsListener(NULL);
        WirelessWAN::SetIndicationsState(false);
        delete myReadyStateListener;

        return false;
    }

    // Check the ready state flag
    readyState = wwanReadyStatePtr->State;
    free(wwanReadyStatePtr); // Memory was allocated by "GetReadyState(...)"
    if ((readyState & WWAN_READY_STATE_INITIALIZED) != 0 )
    {

        // Already in the ready state.
        // Remove indication registration and cleanup objects
        WirelessWAN::SetIndicationsListener(NULL);
        WirelessWAN::SetIndicationsState(false);
        delete myReadyStateListener;

        return true;
```

Example 22.1 Initializing WWAN in C++ (Continued)

```
    }

    // Wait up to 30 seconds for the listener object to signal ready.
    const int thirtySeconds = 30000;
    DWORD eventWait =
        WaitForSingleObject(myReadyStateListener->GetWwanReadyEvent(),
            thirtySeconds);

    // Remove indication registration and cleanup objects
    WirelessWAN::SetIndicationsListener(NULL);
    WirelessWAN::SetIndicationsState(false);
    delete myReadyStateListener;

    if (eventWait == WAIT_TIMEOUT)
    {

        // Ready event not received within 30 seconds;
        return false;
    }

    // The ready event was received.
    return true;
}
```

This example shows how to initialize WWAN without setting the indications listener.

Example 22.2 Initializing WWAN in C#

```
private bool InitWWANandWait()
{
    WWanRequestParameters wwanParameters = new WWanRequestParameters();
    WWAN_READY_STATE wwanReadyStatePtr   = new WWAN_READY_STATE();

    WwanRequestStatus retVal = WirelessWAN.GetReadyState(wwanParameters,
        wwanReadyStatePtr);

    if (retVal == WwanRequestStatus.NotInitialized)
    {
        if (WirelessWAN.Initialize() != WwanRequestStatus.Success)
        {
            return false;
        }
    }

    int count = 60;
    while (count > 0)
    {
        Thread.Sleep(1000);

        // Get the ready state again....
        retVal = WirelessWAN.GetReadyState(wwanParameters, wwanReadyStatePtr);

        if (
            (retVal != WwanRequestStatus.Success)               ||
            (wwanParameters.resultCode != WwanResult.Success)   ||
            (wwanReadyStatePtr == null)
           )
        {
            // Unable to get the ready state.  See return value / result code for re
ason.
            return false;
        }
```

Example 22.2 Initializing WWAN in C# (Continued)

```
        if ((wwanReadyStatePtr.State & WWAN_READY_STATE_FLAGS.INITIALIZED) != 0)
        {
            // Already in the ready state.
            return true;
        }

        count--;
    }
    return false;
}
```

22.5 Closing WWAN

When an application has finished using the WWAN interface, the WWAN interface **must** be closed. It is essential that the interface is closed before the application exits.

Warning: If a Shutdown call is not paired with each Initialize call, system resources are not released or cleaned up properly, resulting in a memory leak.

Example 22.3 Closing WWAN in C++

```
// For the initialization process
// see Example 22.1  Initializing WWAN in C++ on page 202
if (InitializeWWANAndWait())
{
    // Do WWAN work here...

    // Shutdown the interface
    WirelessWAN::Shutdown();
}
```

Example 22.4 Closing WWAN in C#

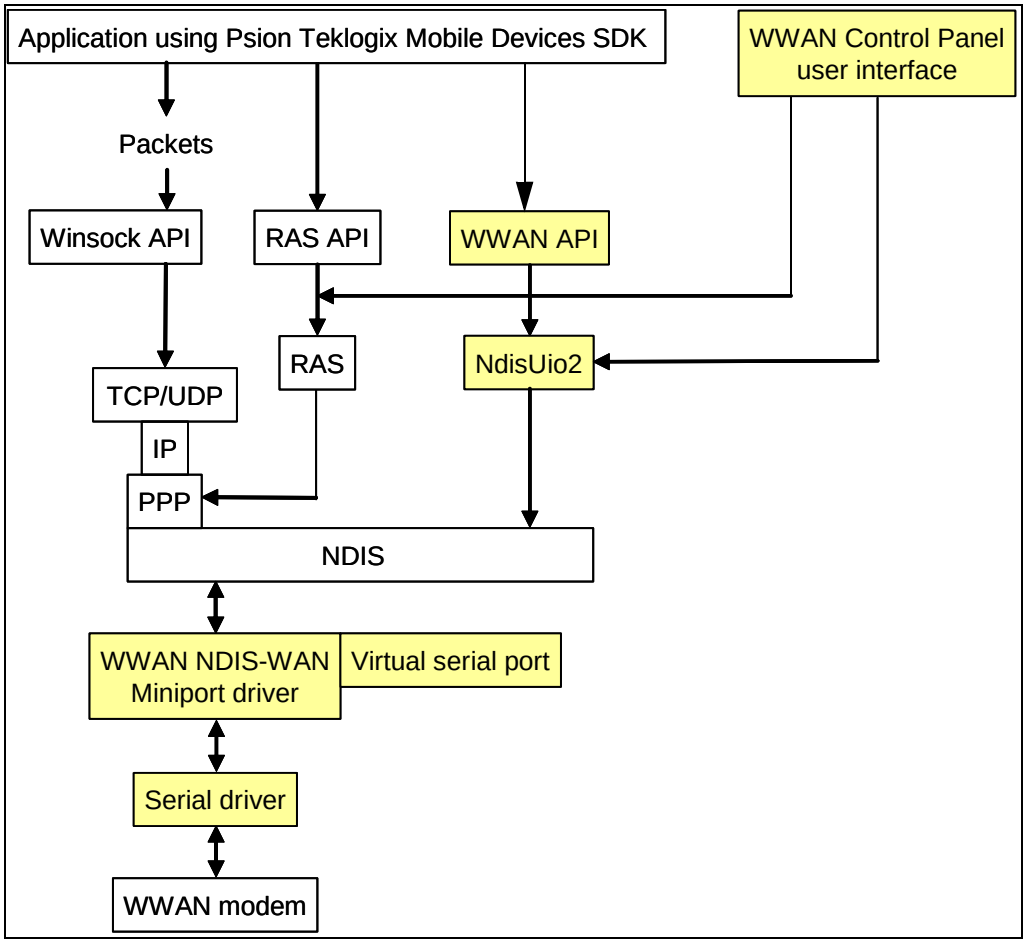
```
// For the initialization process see Example 22.2  Initializing WWAN in C# on
page 204
if (InitializeWWANAndWait())
{
    // Do WWAN work here...

    // Shutdown (clean up) the WWAN interface
    WirelessWAN.Shutdown();
}
```

22.6 Connecting To The Internet

A WWAN modem must initiate a connection. This behaviour is similar to a dial-up connection but different from an Ethernet interface which connects automatically. When a WWAN modem is attached to a network and powered up, it detects the network, but is unable to transfer packets to the network until a packet data connection is set up.

Figure 22.1 Interaction Between Psion Teklogix WWAN And Windows Components



The WWAN functions in the Mobile Devices SDK provide extra features that are needed by a WWAN connection, but not by an Ethernet or dial-up connection. These features include the ability to:

- Query the radio signal strength
- Select one of several available networks

The Wireless Manager applet in the Control Panel allows you to configure settings for the WWAN connection. This gives access to many of the WWAN driver functions, without the necessity of writing code. This includes activities such as querying the signal strength. The Wireless Manager applet uses RAS to connect and disconnect WWAN connections.

22.6.1 Windows CE 5.0 Connection

Windows CE 5.0 provides the RAS (Remote Access Service) API for making Internet connections. An application can use the Windows RAS API to set up and tear down a WWAN connection.

Note: You must initialize WWAN before making the connection. See "Initializing WWAN" on page 199 for details.

See also "The DbGprs.csv File" on page 209 for information that may need to be provided to complete the connection.

*Note: It is recommended to wait for a Packet Service Status (see Section 22.6.4 on page 210) of **Wwan AttachStateAttached** before attempting a connection through RAS.*

The following example uses **RasGetEntryDialParams** and **RasDial** to start a connection.

Example 22.5 Windows CE 5.0 Internet Connection in C++

```
#include <wwan.hpp>
#include <ras.h>

using namespace PsionTeklogix::WWAN;

//
// WWANRas_Connect()
//
//  This function is called in order to establish a RAS connection through the WWAN
//  driver.
//
// Parameters:
//  entryName   Name of the RAS phone book entry.  For WWAN, this should always be
//              "Wirless WAN"
//  password    Password corresponding (if none was stored earlier)
//  hWnd        Handle of window that will receive connection state messages.  You
//              receive this via the WM_RASDIALEVENT windows message.
//
// Notes:
//  REMEMBER: Initialize WWAN interface first!
//
// Returns:
//  TRUE for success, FALSE otherwise
//
//
BOOL WWANRas_Connect(WCHAR const * entryName,
              WCHAR const * username,
              WCHAR const * password,
              HWND          hWnd)
{
    HRASCONN         hRasConn;
    RASDIALPARAMS   rasDialParams;
    BOOL            passwordSet;

    // Initialize the RASDIALPARAMS structure.
    // NOTE: When connecting to WWAN, entryName will be "Wireless WAN".
    memset (&rasDialParams, 0, sizeof (RASDIALPARAMS));
    rasDialParams.dwSize = sizeof (RASDIALPARAMS);
    wcsncpy (rasDialParams.szEntryName, entryName, RAS_MaxEntryName);

    if (RasGetEntryDialParams(NULL,
                              &rasDialParams,
                              &passwordSet) != 0)
    {
        return FALSE;
    }
    // If the password was not automatically determined, then provide it here.
    if (!passwordSet)
    {
    wcscpy (rasDialParams.szUserName, username);
        wcscpy (rasDialParams.szPassword, password);
    }

    // Connect
    if (RasDial(NULL, NULL, &rasDialParams, 0xffffffff, hWnd, &hRasConn) != 0)
    {
```

Example 22.5 Windows CE 5.0 Internet Connection in C++ (Continued)

```
        if (hRasComm != NULL)
        {
            RasHangUp(hRasComm);
            hRasComm = NULL;
        }
        return FALSE;
    }
    return TRUE;
}

//
// Window's Message handler (WndProc)
//
case WM_RASDIALEVENT:
{
    RASCONNSTATE rasconnstate = (RASCONNSTATE) wParam;
    DWORD        dwError      = (DWORD) lParam;
}
return TRUE;
```

The following example uses the RAS namespace for the Mobile Devices SDK to make the Internet connection. See Chapter 12: “RAS (Remote Access Service)” for more information.

Example 22.6 Windows CE 5.0 Internet Connection in C#

```
using PsionTeklogix.WWAN;
using PsionTeklogix.RAS;

bool returnValue = true;

// see ealier section on WWAN Initialization
if (InitWWANandWait())
{
    private Entry wwanEntry = new Entry("Wireless WAN");

wwanEntry.UserName = "ISP@CINGULARGPRS.COM";
wwanEntry.Password = "CINGULAR1";

try
{
               if (!wwanEntry.Dial(false))
           {
        returnValue = false;
           }
         }
         catch (Exception ex)
         {
            MessageBox.Show("FAIL: \r\n" + ex.Message, "RAS Dial");
        returnValue = false;
         }
}
return returnValue;
```

22.6.2 Windows Mobile 5.0 And Windows Mobile 6 Connection

For Windows Mobile operating systems, Microsoft provides the Connection Manager API. This replaces RAS for making WWAN connections.

For more information on using the Connection Manager API, visit the following URLs:

http://msdn2.microsoft.com/en-us/library/bb416435.aspx
http://msdn2.microsoft.com/en-us/library/bb840031.aspx
http://msdn.microsoft.com/en-ca/netframework/dd296752.aspx

See also "The DbGprs.csv File" on page 209 for information that may need to be provided to complete the connection.

*Note: It is recommended to wait for a Packet Service Status (see Section 22.6.4) of **Wwan AttachStateAttached** before attempting a connection through Connection Manager.*

The following example shows how to connect to the Internet using the Connection Manager:

Example 22.7 Connecting to the Internet using Connection Manager in C#

```
// pinvoke region required to access native Connection
// Manager not included here.
//

const int _syncConnectTimeout = 60000; // 60 seconds
string url = "http://msdn2.microsoft.com";
int ncache = 0;

if (status == ConnMgrStatus.Connected)
   ConnMgrReleaseConnection(_connectionHandle, ncache);

Guid networkGuid = Guid.Empty;

ConnMgrMapURL(url, ref networkGuid, 0);
ConnMgrConnectionInfo info =
new ConnMgrConnectionInfo(networkGuid, ConnMgrPriority.UserInteractive);
ConnMgrEstablishConnectionSync(info, ref _connectionHandle, _syncConnectTimeout, ref
 status);
ConnMgrEstablishConnection(info, ref _connectionHandle);

if (status == ConnMgrStatus.Connected)
{
//
       // Success!
       //
}
```

22.6.3 The DbGprs.csv File

To make an Internet connection using either RAS or the Connection Manager a username, a password, and some other related information—such as APN, DNS, MCC, or MNC—may be needed. Psion Teklogix provides this information in the **...\Windows\DbGprs.csv** file, on every Psion Teklogix computer that supports WWAN.

This file is in comma delimited format, so it can be easily read with a text editor or loaded into a spreadsheet program. When you locate your WWAN service provider in this file, the most common information types required by the provider are listed.

22.6.4 Checking Packet Data Status

If packet data is enabled for the connection to the WWAN service provider, The WWAN driver automatically configures the system so that a connection can be made using RAS or Connection Manager.

The following example shows how to check the packet data status:

Example 22.8 Checking Packet Service Status In C++

```
WWanRequestParameters opts;
opts.hWaitForAsyncCompletion = 0;
WWAN_PACKET_SERVICE packetService;

if (WirelessWAN::GetPacketService(opts, packetService) == WwanRequestSuccess)
{

    if (packetService.AttachState == WwanAttachStateAttached)
    {
        // Packet data is now enabled. WWAN communication can start...
    }
    else if (packetService.AttachState == WwanAttachStateDetached)
    {
        // Packet data is NOT enabled, which means that either
        // the service provider does not have packet data enabled,
        // or the WWAN driver did not automatically select a network
        // at startup, and it must be selected manually here.
    }
}
```

22.6.5 Roaming

Typically, in your home country the home network is selected automatically, and all other networks are forbidden. There is usually no national roaming.

If you are in another country, several networks may have roaming agreements with your home network and you may want to manually select one—for example, the partner with the best tariff.

If the network is not selected manually, the WWAN modem automatically selects the network with the highest signal strength.

If you are in a border area you can manually select your home network, in order to prevent roaming to a partner network across the border with higher roaming charges.

Use **SetRegisterState** to manually select the a WWAN network. See “Access Flags” on page 212 for information on using the Set Register State Access Flag.

22.6.6 Connecting Manually to A WWAN Network

Usually, at startup the WWAN driver selects, connects to, and registers with, the appropriate networks for the SIM card.

If you call **GetRegisterState** and the return value indicates that the connection is *deregistered*, then you must do one of the following to select the network:

- Use **SetRegisterState**, *or*
- Use the WWAN settings in the Wireless Manager Control Panel applet

See “The DbGprs.csv File” on page 209 for information that must be provided to the WWAN network during connection.

22.6.7 Selecting An Access Point Name (APN)

An Access Point Name (APN) identifies an external network that is accessible from a Psion Teklogix computer. An APN has several associated attributes that define how the computer can access the external network at that point.

In most cellular networks there are two types of APNs:

- Wireless Application Protocol (WAP) APN, enabling access to the default WAP content (or the network operator's *walled garden*). This APN normally filters non-WAP content by traffic, ports, or volume.
- Internet or WEB APN, which enables access to all Internet content.This APN is normally unfiltered and often the network operator charges a higher tariff for it. This connection is needed when using applications or an HTML browser.

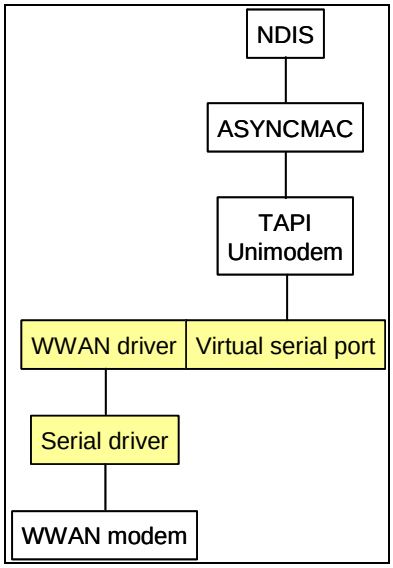
By default, the user interface (WWANUI) sets up the correct packet data context., including the Internet APN. You may want to configure more APNs, if you have arranged with your service provider to use more than one SIM card, or to grant your SIM card access to several APNs. Use **SetProvisionedContexts** to make these APN changes.

User name, password, and authentication method have to be set through RAS; however, to be compatible with possible future SDK changes, applications should fill in *AuthenticationType*, *UserName*, and *Password* when calling **SetProvisionedContexts**.

22.6.8 Connecting To A VPN

A VPN (Virtual Private Network) can be used to connect a WWAN modem to a corporate network. Some third-party VPN clients work correctly only in conjunction with the Microsoft ASYNCMAC driver. In the standard Mobile Devices SDK implementation of WWAN, ASYNCMAC is not used. Psion Teklogix proprietary modules complete the communication between NDIS and the WWAN modem.

When using a third-party VPN client, in the Wireless Manager applet WWAN configuration screen, select **Use virtual serial port**. This gives access to ASYNCMAC through the virtual serial port. This is shown in the following diagram:



The virtual serial port cannot be enabled from within the Mobile Devices SDK.

22.7 Access Flags

The Wireless Manager Control Panel applet and the Mobile Devices SDK are able to read and set the same WWAN settings. The access flags prevent conflicts between values manipulated by the Wireless Manager interface and by the Mobile Devices SDK. Most of the WWAN settings can be queried through both access methods; however, the following four activities can be carried out by both access methods, and should be carried out by an SDK application only after setting the corresponding access flag:

- Set the network register state
- Set the PIN
- Set provisioned contexts
- Set suppress call notifications

The Mobile Devices SDK can set an access flag corresponding to each of these settings. While the access flag is set, the Wireless Manager applet user interface cannot change the corresponding setting. While one of these access flags is enabled, the application must handle everything associated with the associated activity: The user interface is locked out until the access flag is disabled.

Set Register State Access Flag

When GSM radio modem is roaming, there are two courses of action. The first option is that either the user or the software application selects a network. The second option is that the radio automatically selects a network. An SDK application that explicitly selects a network must set this flag.

When the set register access flag is enabled, the user is prevented from selecting the network though the Control Panel. The **Tools** -> **Network** menu is not available in the Wireless Manager applet WWAN UI settings screen.

If this flag is set during driver initialization, and the startup registration mode for the modem is **WwanRegisterModeDeregister**, then the user interface does not automatically select a network at startup—this is the case for WORKABOUT PRO expansion boards with Chi Mei modems. In this case, the network must be explicitly selected through the user interface or via the application.

Set PIN Access Flag

When the **Set PIN access flag** is enabled, the PIN cannot be entered through the Control Panel.

If this flag is set during driver initialization, then the user interface does not prompt the user for a PIN or a PUK. The initialization progress status remains at **Initializing modem** until the application that set the flag handles the PIN initialization.

Set Provisioned Contexts Access Flag

If this flag is set during driver initialization, the user interface is not used to set up the data configuration profile for the driver. As a result, even though the status is shown as **Ready to connect**, the [CONNECT DATA] button is not available. The application that set this flag is responsible for setting up the data configuration before establishing a packet data connection through RAS.

Set Suppress Call Notifications Access Flag

When this flag is set, the WWAN Control Panel program does not notify the user of the Psion Teklogix computer when there is an incoming call. When this flag is not set, the WWAN program always notifies the user of incoming calls—see the user manual for your Psion Teklogix computer for further details.

This flag works with version 1.10047 and later of the WWANUI Control Panel program, accessed through the Wireless Manager applet.

22.7.1 Controlling The Interaction With The Control Panel

The following code samples show how the access flags are manipulated using the Mobile Devices SDK. The process is as follows:

1. Save the current access flag settings
2. Change the access flag settings

Example 22.9 Suppressing Voice Notifications in C++

```
DWORD currentAccessFlags = 0;
WirelessWAN::GetCurrentAccessFlags(currentAccessFlags);
currentAccessFlags |= AccessFlags_SuppressCallNotifications;
WirelessWAN::SetCurrentAccessFlags(currentAccessFlags);
```

Example 22.10 Suppressing Voice Notifications in C#

```
AccessFlags accessFlags = 0;
if (WirelessWAN.GetCurrentAccessFlags(ref accessFlags) == WwanRequestStatus.Success)
{
    accessFlags |= AccessFlags.SuppressCallNotifications;
    if (WirelessWAN.SetCurrentAccessFlags(accessFlags) != WwanRequestStatus.Success)
    {

        // Failure!
    }
}
```

Warning: Before the application closes, control of the access flags must be returned to the WWAN Control Panel program.

Example 22.11 Returning Control Of The Access Flags To The Control Panel in C++

```
// Re-enable the Control Panel notifications:
DWORD currentAccessFlags = 0;
WirelessWAN::GetCurrentAccessFlags(currentAccessFlags);
currentAccessFlags &= ~AccessFlags_SuppressCallNotifications;
WirelessWAN::SetCurrentAccessFlags(currentAccessFlags);
```

22.8 RSSI (Received Signal Strength Indicator)

The RSSI status of the radio changes continuously, due to various environmental factors. There is no **get** function for RSSI in the Mobile Devices SDK. The signal strength can only be obtained by registering a callback function. A **SetSignalSuggestion** feature is implemented which indicates to the driver how often the signal strength should be measured. Often signal strength is reported using a bar graph. A callback is only necessary when the increment to the next graphed level is reached. Callbacks prevent excessive activity on the processor and the modem.

The signal strength can only be obtained by registering a callback function, **OnSignalStrength** in the **IndicationsListener** class, as shown here:

Example 22.12 Getting The signal Strength in C++

```
class MyRssiListener : public PsionTeklogix::WWAN::IndicationsListener
{
public:
    MyRssiListener()
    {
    }
```

Example 22.12 Getting The signal Strength in C++ (Continued)

```
    ~MyRssiListener()
    {
    }

    virtual void OnSignalState(const WWAN_SIGNAL_STATE &signalState)
    {
    //
    // Process RSSI here
    //
    }
};

//
// Later in the code.
//

MyRssiListener *rssi = new MyRssiListener();
WirelessWAN::SetIndicationsListener(rssi);
WirelessWAN::SetIndicationsState(true);
```

Fore more refined control over when the RSSI signal strength is measured, use the **SetSignalSuggestion**. This causes the WWAN driver to invoke **OnSignalStrength** whenever the signal strength changes by more than the *RssiThreshold* value. The *RssiInterval* (in seconds) value determines how often the WWAN driver queries the modem for the signal strength. This is shown in the following:

Example 22.13 Using SetSignalSuggestion in C++

```
WWanRequestParameters params;
params.hWaitForAsyncCompletion = 0;
WWAN_SIGNAL_STATE signalState;
signalState.RssiInterval  = 10;  // 10 second interval
signalState.RssiThreshold = 2;   // 2 dbm change

if (WirelessWAN::SetSignalSuggestion(params, signalState) == WwanRequestSuccess)
{
if( params.resultCode == WwanResultSuccess )
    {
            // Settings successfully changed.  You'll now start getting
            // signal strength indications via OnSignalStrength
    }
}
```

22.9 WWAN API Elements

C++: WWAN on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix::WWAN** namespace.

Java: Not available.

.NET: WWAN on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix.WWAN** namespace.

22.10 Using SMS (Short Message Service)

SMS (Short Message Service) is available on GSM networks. It allows text messages of up to 160 characters to be sent and received through the network operator's SMS gateway. If the receiver is powered off or out of range, messages are stored in the network and delivered at the first opportunity.

On Psion Teklogix systems, SMS messages are stored on a SIM card, or in the WWAN modem. At startup, initialization of the SIM card takes more time than initialization of the modem.

When callbacks are used to track initialization, typically the application is notified of the ready status of the modem, and then some time later it is notified of the ready status of the SIM card. Only then are the SMS functions available. Some modems do not support SMS; in these cases the SMS ready status never occurs.

Incoming SMS messages are stored on one or several mailboxes. The mailbox is a logical entity. The mailboxes are numbered sequentially—the mailbox index—starting from 0 (zero) for the first mailbox. Within each mailbox the messages are numbered sequentially—the message index.

Deleting a message can cause the mailbox to have gaps in its sequence of message indices (for example, deleting message #2 from a mailbox containing messages at indices 1, 2, and 3 results in messages at indices 1 and 3). An application must take this into account when retrieving messages. Functions are provided for obtaining the number of mailboxes, the current message pointer, and the maximum numbers of messages in each mailbox.

Some mailboxes are writable, this means that sent messages can be stored in the mailbox.

When SMS functions are called, an SMS location structure is passed in as a parameter. The SMS location structure holds the mailbox index and the message index within the mailbox.

If a callback is registered for incoming SMS messages, a location structure is returned by the callback that specifies the mailbox index and the message index. This location structure can be passed to subsequent SMS calls that process the incoming SMS message. It is recommended that applications query the location of each message, and not make assumptions about where the modem put the message.

Most modems have only one mailbox that is stored on the SIM card. A few modems have a second mailbox that is stored in the modem itself. The Mobile Devices SDK provides information on the number and structure of the mailboxes. Normally, the modem automatically selects the mailbox for an incoming message. When one mailbox is full, the modem starts filling the next one. To make mailboxes portable between hand-held computers, an application can force a mailbox to be on the SIM card.

22.10.1 SMS API Elements

C++: SMS on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix::WWAN** namespace.

Java: Not available.

.NET: SMS on all Psion Teklogix Windows CE computers is controlled using the **PsionTeklogix.WWAN** namespace.

22.11 WWAN Supplementary Services

Some WWAN networks support supplementary services.These services can complement both voice calls and data services. Setting and querying each supplementary service takes between 30 and 60 seconds, as the Psion Teklogix computer must contact the WWAN network and wait for a reply. The Mobile Devices SDK provides access to the following supplementary services, if they are supported by your WWAN network:

Incall Supplementary Services

Call Hold

This service enables the subscriber to interrupt an ongoing call and then subsequently reestablish the call. The call hold service is only applicable to normal telephony.

Call Waiting

This service enables the mobile subscriber to be notified of an incoming call during a call. The subscriber can answer, reject, or ignore the incoming call. Call waiting is applicable to all GSM telecommunications services using a circuit-switched connection.

Multiparty Service (Conference Calls)

The multiparty service enables a mobile subscriber to establish a multiparty conversation—that is, a simultaneous conversation between from three to six subscribers. This service is only applicable to normal telephony.

Explicit Call Transfer

This service allows a subscriber who has two calls to connect the other parties in those calls, and at the same time disconnect themselves from the call. This service is only supported by the HC25 modem.

Other Supplementary Services

Call Forwarding

This service gives the subscriber the ability to forward incoming calls to another number if the called mobile unit is not reachable, if it is busy, if there is no reply, or if call forwarding is allowed unconditionally.

Calling Line Identification Presentation/Restriction (Caller ID)

These services supply the called party with the integrated services digital network (ISDN) number of the calling party. The restriction service enables the calling party to prevent the presentation of the number. The restriction overrides the presentation.

Call Barring

This service allows a subscriber to restrict selected types of calls. Typically these calls are: all incoming calls; all incoming calls when roaming; all outgoing calls; or, all outgoing international calls.

This service is not supported by the Mobile Devices SDK; however, it is available through the Control Panel.

22.11.1 Voice Service States

These states can apply to several of the supplementary services; however, in this implementation, only call forwarding, call waiting and call barring use them.

Enabled

The service is fully enabled.

Disabled

The service is fully disabled.

Some Enabled

Some of the service is enabled. For example, if call forwarding is enabled only when the line is busy and disabled for all other reasons; then when the status for *All Reasons* is requested, *Some Enabled* is returned.

Unknown

Unable to retrieve the status of the service.

22.12 Voice Calls On A WWAN

Voice calls are supported by the following computers:

- WORKABOUT PRO G2
- Ikôn

Windows CE 5.0

For this version of Windows, the Mobile Devices SDK supports voice call on a WWAN for WORKABOUT PRO G2 computers only.

Windows Mobile 6 Professional

Windows Mobile 6 Professional provides support for voice calls on WWAN; see MSDN for more information. (http://msdn.microsoft.com/en-us/library/bb416512.aspx)

The Mobile Devices SDK does not support voice call for Windows Mobile 6 Professional.

22.12.1 Initializing WWAN

Before initializing voice over WWAN, initialize the WWAN connection following the instructions in "Initializing WWAN" on page 199.

22.12.2 Initializing Voice Over WWAN

Follow these steps to initialize voice over WWAN:

1. Instantiate an instance of the **VoiceCallManager** class —this class controls most of the voice call functions.
2. Create a **VoiceCallManager Listener** object—this makes **OnIncomingCallEvent** available to handle all incoming calls.
3. Use **RegisterListener** to install and register the listener.
4. If the application—not the Control Panel applet—is to handle notifications for incoming voice calls, set the suppress call notifications access flag. See "Access Flags" on page 212 for details.

The following sample code shows how to do this:

Example 22.14 Initializing Voice Over WWAN In C++

```
using namespace PsionTeklogix::WWAN;

class MyVoiceCallManagerListener : public VoiceCallManager::Listener
{
public:

     // OnIncomingCallEvent
     //
     // Description:
     //    This method is invoked by a VoiceCallManager object when there is an inco
ming
     //    voice call (including waiting calls). In this implementation, all new Voi
ceCalls
     //    are added to the voiceSharedData.myVoiceCalls list. A MessageBox is then
     //    displayed asking the user if the call should be answered.  The new status
 is
     //    displayed in the event window in the dialog.
     //
     // Parameters:
     //    incomingCall    - The new incoming call.
     //
    void OnIncomingCallEvent(VoiceCall incomingCall)
    {
       // Add functions for acting on incoming calls here....
    }
};

class CallListener : public VoiceCall::Listener
{

public:

    void OnStatusChangedEvent(VoiceCall         &changedCall,
                VoiceCallStatus    newStatus,
                VoiceCallFlags        newFlags)
    {
        switch(newStatus)
        {
            case VoiceCallStatus_Active:
            break;
                case VoiceCallStatus_Held:
            break;
        case VoiceCallStatus_Dialing:
            break;
        case VoiceCallStatus_Alerting:
            break;
        case VoiceCallStatus_Incoming:
            break;
        case VoiceCallStatus_Waiting:
            break;
        case VoiceCallStatus_NoAnswer:
            break;
        case VoiceCallStatus_NoCarrier:
            break;
        case VoiceCallStatus_Busy:
            break;
        case VoiceCallStatus_SupplementaryAccepted:
            break;
        case VoiceCallStatus_NormalTermination:
            break;
```

Example 22.14 Initializing Voice Over WWAN In C++ (Continued)

```
                default:
            break;
        }
        if ( (newFlags & VoiceCallFlags_LastEvent) != 0 )
        {

            // This voice call is finished.  Its now safe to deallocate
         // the VoiceCall object, instantiated earlier and referenced
            // by changedCall).
        //
 }
    }
};

static VoiceCallManager        *callManager  = NULL;
static MyVoiceCallManagerListener    *incomingCall = NULL;
static CallListener            *callStatus   = NULL;

bool InitWwanVoiceApi()
{
    // Initialize the WWAN namespace and wait for the driver/modem to be ready.
    // (see previous example of this function at the beginning of WWAN API
    // documentation).
    if(InitializeWWANAndWait())
    {
// Create an instance of the VoiceCallManager object
        callManager = new VoiceCallManager();
    // Create the listener
        incomingCall = new MyVoiceCallManagerListener();
    // Register for call manager events (Namely incoming call notification).
    callManager->RegisterListener(*incomingCall);
    // Create a listener for phone calls (general).
    callStatus = new CallListener();

        // The PsionTeklogix WWAN UI application will generate notifications to
        // the user about incoming voice calls.  To suppress the notifications
        // the following access flag should be set.  (works on WWAN UI versions
        // 1.10047 and later)
        DWORD currentAccessFlags = 0;
        WirelessWAN::GetCurrentAccessFlags(currentAccessFlags);
        currentAccessFlags |= AccessFlags_SuppressCallNotifications;
        WirelessWAN::SetCurrentAccessFlags(currentAccessFlags);

    return true;
    }
    return false;
}
```

22.12.3 Closing Voice Over WWAN

The following steps close voice over WWAN:

1. Free all allocated resources.
2. If the application has been controlling the call notifications, return control to the Control Panel applet. See "Access Flags" on page 212 for details.
3. Call **Shutdown**.

Example 22.15 Closing Voice Over WWAN in C++

```
void DeInitWwanVoiceApi()
{
    if (callManager != NULL)
    {
        // make sure there are no active voice calls since you will lose
        // control of them after deleting the objects.
        callManager->HangUpAll();

        delete callManager;
        callManager = NULL;
    }

    // Delete the VoiceCallManager object before the listener, or deregister
    // the listener object first (just in case an event is generated).
    if (incomingCall != NULL)
    {
        delete incomingCall;
        incomingCall = NULL;
    }

    // Re-enable the WWAN UI notifications:
    DWORD currentAccessFlags = 0;
    WirelessWAN::GetCurrentAccessFlags(currentAccessFlags);
    currentAccessFlags &= ~AccessFlags_SuppressCallNotifications;
    WirelessWAN::SetCurrentAccessFlags(currentAccessFlags);

    // Shutdown WWAN namespace (assuming no other code is using it)
    WirelessWAN::Shutdown();
}
```

22.12.4 Making A Phone Call

Use **DialNumber** to initiate a phone call.

Example 22.16 Dialing a Number In C++

```
// NOTE: GetComboEntry is NOT part of the Mobile Devices SDK WWAN API, but is a
// fictional function designed to get a Phonebook entry object
// from a fictional combo box.

PhonebookEntry *entry = GetComboEntry();
VoiceCall *newCall = new VoiceCall(VoiceCall::DialNumber(entry->GetPhoneNumber()));
newCall->RegisterListener(*callStatus);
```

DialNumber returns a **VoiceCall** object in dialing state. Continue as follows:

1. Create a new object, to ensure persistence, by passing this **VoiceCall** object to a copy constructor. This is necessary if this call is within a Windows procedure.
2. Immediately, use **RegisterListener** to register a listener to monitor the state of the phone call. See "Voice Call States" on page 222 for definitions.

Example 22.17 C++ Voice Call Listener

```
class CallListener : public VoiceCall::Listener
{

public:

    void OnStatusChangedEvent(VoiceCall         &changedCall,
                VoiceCallStatus    newStatus,
                VoiceCallFlags        newFlags)
    {
        switch(newStatus)
        {
            case VoiceCallStatus_Active:
            break;
                case VoiceCallStatus_Held:
            break;
        case VoiceCallStatus_Dialing:
            break;
        case VoiceCallStatus_Alerting:
            break;
        case VoiceCallStatus_Incoming:
            break;
        case VoiceCallStatus_Waiting:
            break;
        case VoiceCallStatus_NoAnswer:
            break;
        case VoiceCallStatus_NoCarrier:
            break;
        case VoiceCallStatus_Busy:
            break;
        case VoiceCallStatus_SupplementaryAccepted:
            break;
        case VoiceCallStatus_NormalTermination:
            break;
                default:
            break;
        }
        if ( (newFlags & VoiceCallFlags_LastEvent) != 0 )
        {
            //
            // This voice call is finished.  Its now safe to deallocate
         // the VoiceCall object you instantiated earlier ( referenced
            // by changedCall).
        //
 }
    }
};
```

22.12.5 Voice Call States

These voice call states describe the state of a voice connection between the Psion Teklogix computer and the WWAN network. Contact your WWAN network provider to determine which of these services they support.

The status can change at any time for reasons that are outside the control of the application. Instead of polling this method, it is recommended that an application creates an event handler.

A typical outbound voice call passes through several of these states, for example:

1. Dialing
2. Alerting
3. Active
4. NormalTermination

An application may not detect all the states that the call passes through.

When the voice call is terminated, the last **Voicecall** event received has its voice call flag set to *last event*. Once this event is received, free the **VoiceCall** object that you created immediately after invoking **DialNumber**. See "Making A Phone Call" on page 220.

A typical inbound call passes through several states, for example:

1. Incoming
2. Active
3. Normal Termination

The event and the flags have different names in C++ and C#.

Voice call on Psion Teklogix computers use the GSM standard. This standard allows a subscriber to have the following calls, at the same time:

- One active call
- One held call
- One waiting call

Either the active call or the held call can be a multiparty call.

Active

Applies to: incoming and outgoing calls

The Psion Teklogix computer is connected to another party over the WWAN network: end-to-end conversation is possible.

Held

Applies to: incoming and outgoing calls

Only available if the WWAN provider supports this feature.

The Psion Teklogix computer has temporarily interrupted the voice call. End-to-end conversation is not possible. The conversation can be reestablished later.

Dialing

Applies to: outgoing calls

The WWAN network is in the process of contacting the called party.

Alerting

Applies to: outgoing calls

The called party is being notified that there is an incoming voice call. Typically this means that the called phone is ringing.

Incoming

Applies to: incoming calls

This state applies to an incoming voice call when there is no active voice call. See "Audio for Voice Over WWAN" on page 228 for instructions on creating a ring tone.

Waiting

Applies to: incoming calls

Only available if the WWAN provider supports this feature.

This state applies to an incoming call when there is an active voice call, or a previous call in the incoming state.

No Answer

Applies to: outgoing calls

The called party did not answer the call within a specified, network defined, time.

This state is not often returned as many calls are automatically answered by switchboards and answering systems.

No Carrier

Applies to: outgoing calls

The Psion Teklogix computer could not connect to the WWAN network.

Busy

Applies to: outgoing calls

The called party is busy and does not answer the call.

This state is not often returned as many calls are automatically answered by switchboards and answering systems.

Supplementary Accepted

Applies to: outgoing calls

Supplementary services can be configured using codes entered on the keypad of a telephone. These codes are numerals preceded by *#. An application can insert these codes into a dialing string. This status indicates that the network accepted one of these codes.

For a list of supplementary services see "WWAN Supplementary Services" on page 216. For a list of codes, consult your network provider.

Normal Termination

Applies to: incoming and outgoing calls

The voice call terminated. This is returned by any call where the modem detects that a call, that has been active, no longer exists.

22.12.6 Receiving A Phone Call

The **VoiceCallManager** object is responsible for answering incoming phone calls through the listener that was registered during initialization—see "Initializing Voice Over WWAN" on page 217 for details. An incoming call generates an incoming call event that is processed by the **VoiceCallManager** object.

Terminology is different in C++ and C#.

The following example shows how to receive incoming phone calls:

1. Play a ring tone.
2. Ask if the user wants to answer the call—includes displaying the caller ID of the incoming phone call.
3. Either activate the incoming call or reject it.

Example 22.18 Answering a Phone Call in C++

```
class MyVoiceCallManagerListener : public VoiceCallManager::Listener
{
public:

     // OnIncomingCallEvent
     //
     // Description:
     //  This method is invoked by a VoiceCallManager object when there is an incomi
ng
     //  voice call (including waiting calls).  A MessageBox is then displayed askin
g the
     //  user if the call should be answered.  The new status is displayed in the ev
ent
     //  window in the dialog.
     //
     // Parameters:
     //  incomingCall    - The new incoming call.

    void OnIncomingCallEvent(VoiceCall incomingCall)
    {
    //
    // Instantiate a new VoiceCall object, and register a listner.
    // The listener is responsible for determining when/how to
    // free up this memory.
    //
    VoiceCall *newCall = new VoiceCall(incomingCall);
    newCall->RegisterListener(*callStatus);

        try
        {
            try
            {
                // Play a ring tone to notify the user of an incoming call.
                // From PsionTeklogix::Sound namespace.
                PlayRepeatingWave(std::wstring(L"\\windows\\ring.wav"), 3000 );
            }
            catch(std::runtime_error ex)
            {
                /* unable to play the ringtone, does the file exist? */
            }

            // Ask the user if they want to answer the call.
            WCHAR messageBuffer[1024];
            wsprintf(messageBuffer,
        L"Answer incoming call from: \"%s\"?\r\nPress NO to reject the call.",
```

Example 22.18 Answering a Phone Call in C++ (Continued)

```
          incomingCall.GetOtherPartyCallerId().c_str());

 // Please note: It is not recomended that you block this thread,
            // the next event will not be received untill this handler returns.
            // This blocking message box is here to keep this demo app simple.
            int rc = MessageBox(0, messageBuffer, L"Incoming Call",
                (MB_YESNOCANCEL | MB_TOPMOST | MB_SETFOREGROUND));

            // The user responded, don't need to play the ring tone any more.
            // From the PsionTeklogix::Sound namespace.
            StopRepeatingWave();

            if (rc == IDYES)
            {
        // answer the incoming call
                incomingCall.AnswerIncoming();
            }
         else if (rc == IDNO)
            {
        // reject the incoming call
                incomingCall.RejectIncoming();
            }
         else
            {
                // call was ignored
            }
        }
        catch(std::runtime_error&)
        {
            // If we are here, the call was probably terminated before the user resp
onded
            // to the messagebox, the object is no longer valid.
        }
    }
};
```

22.12.7 Processing Voice Calls

This section outlines the voice call states.

Incoming Call

An incoming call generates a **VoiceCallManager** incoming call event with a voice call state of *Incoming* if it is the **only** call being received.

Call Waiting

Call waiting must be available from the WWAN network provider and enabled on the Psion Teklogix computer before an incoming call can be placed in this state. Use **GetCallWaitingEnabled** to check this status. If call waiting is not already enabled, use **SetCallWaitingEnabled** to enable call waiting.

An incoming call generates a **VoiceCallManager** incoming call event with a voice call state of *Call waiting* if another call is in the voice call state of *Active*. There can be only one call at a time in the call waiting state.

Placing The Active Call On Hold And Answering The Waiting Call

If there is an active call, use **AnswerIncoming** to place the active call on hold, and to answer the waiting call.

If there is one or more calls on hold, then the active call joins them on hold. The held calls can converse in a multiparty call excluding you. The incoming call becomes the new active call. If you do not want this to happen either the active call or the held call must be disconnected before **AnswerIncoming** is called.

Placing The Active Call On Hold And Answering The Held, Or The Waiting, Call

Use **SwapHeldAndActive**. This function does one of the following:

- Places the active call on hold, and makes the held call active, *or*
- Places the active call on hold, and makes the waiting call active.

This function fails if there is both a held call and a waiting call.

Holding Calls

The Mobile Devices SDK supports holding calls. Although holding calls are only available when the Psion Teklogix computer is associated with a WWAN network that provides a held calls service.

A held call is a call that has been active, but is temporarily interrupted. It can be resumed, made active, again.

Multiparty Calls (Conference Calls)

The Mobile Devices SDK supports multiparty calls. Although multiparty calls are only available when the Psion Teklogix computer is associated with a WWAN network that provides a multi-party calling service. Up to five calls can be joined into a conference call.

Use the **VoiceCallManager** function **RetrieveHeld** to join together the active call with all the held calls. All the calls are now active, and all the parties can communicate with each other.

The behaviour of **RetrieveHeld** depends on the WWAN modem. For example, this call is successful with the MC75 modem when there is one held call and no active calls, while it fails with the HC25 modem if there is not at least one active call and at least one held call.

Adding A Call To A Multiparty Call

If an incoming call event is received, with a voice call state of *Call waiting*, use the **VoiceCall** function **ActivateAndHoldOthers** to make this the active call and place all the other calls on hold. This function fails if there is already a call on hold.

Use the **VoiceCallManager** function **RetrieveHeld** to join together the active call with all the held calls. All parties, including the new call, can now communicate with each other.

Use the **VoiceCallManager** function **HoldActive** to place all the active calls on hold. Then one of the following occurs:

- If there is a waiting call, it becomes the active call, *or*
- If there is a call on hold, but there is no waiting call, the held call becomes the active call, *or*
- If there is both a held call and a waiting call, the command is rejected.

22.12.8 Terminating Voice Calls

Rejecting An Incoming Call

While processing the incoming call event, use **RejectIncoming** to reject the call. See Example 22.18 Answering a Phone Call in C++ on page 224.

Terminating Active Calls

Terminating One Active Call

Use **HangUp** or **HangUpActive** to terminate a single active call.

Terminating All Active Calls

Use **HangUpActive** to terminate all active calls.

If there is a call in the call waiting state, it becomes active after the active calls are terminated.

If there is no call in the call waiting state, then the held calls becomes active.

Terminating All Held Calls

Use **HangUpHeld** to terminate all held calls.

If there is a waiting call, this function fails. So, use **RejectIncoming** on all the waiting call before calling **HangUpHeld**.

Terminating All Active And Held Calls

Use **HangUpAll** to terminate all calls in the active state and all calls in the held state. If there is a waiting call, it becomes active.

Terminating One Selected Call

Use the **VoiceCall** function **HangUp** to terminate one selected call. The behaviour of **HangUp** depends on the WWAN modem. The MC75 terminates a call in any state. The HC25 terminates calls in the active and held states.

22.12.9 Call Forwarding

The Mobile Devices SDK supports call forwarding. Although call forwarding is only available when the Psion Teklogix computer is associated with a WWAN network that provides a call forwarding service.

Conditional Call Forwarding

Conditional call forwarding has one of the following settings:

- **Busy:** An incoming call is forwarded if there is already an active call.
- **No answer:** An incoming call is forwarded if it is not answered within a specified time.
- **Not reachable:** An incoming call is forwarded when the WWAN network cannot reach the Psion Teklogix computer.

Unconditional Call Forwarding

Unconditional call forwarding forwards all incoming calls. Unconditional forwarding takes precedence over any conditional forwarding settings that are in force at the same time.

22.12.10 Dual-Tone Multifrequency (DTMF)

DTMF is a tone signalling scheme often used for control purposes on a telephone network. For example, it can be used for the remote control of an answering machine, to control voice mail, or to dial an extension.

For example, to dial an extension number of 1234, there are two options:

- Call **SendDTMF** with the string **1234**, *or*
- Call **SendDTMF** four times. The first time set the string to **1**, the second time set the string to **2**, then to **3**, then to **4**.

You can use SendDTMF for any touch-tone service.

22.12.11 Blocking Inbound And Outbound Calls

To ensure that the application has control of the call notifications, set the suppress call notifications flag before blocking calls. See "Access Flags" on page 212 for details.

Blocking (Rejecting) Inbound Calls

While processing the **VoiceCallManager** incoming call event, use **GetOtherPartyCallerId** to get the caller ID. If the caller ID is recognized as one that should be blocked, use **RejectIncoming** to reject the call.

If the incoming call has its caller ID blocked, **GetOtherPartyCallerId** returns an empty string. Also, in the **Voicecall** status changed event, the voice call flag is *withheld incoming number*.

Blocking Outbound Calls

To prevent phone calls to selected phone numbers, monitor the **Voicecall** status changed events. While the outbound call is in the *alerting* state, use **HangUp** to block the call.

22.12.12 Blocking Caller ID On Outgoing Calls

Use **SetOutgoingCallerIdEnabled** to block caller ID on outgoing calls. Use **GetOutgoingCallerIdEnabled** to query the state of the outgoing caller ID.

22.12.13 Audio for Voice Over WWAN

Playing A Ring Tone

Use **PlayRepeatingWave**, in the **Sound** namespace, to play a repeating tone. This is shown in the following example:

Example 22.19 Playing A Repeating Tone in C++

```
try
{
    // Play a ring tone to notify the user of an incoming call.
    // From PsionTeklogix::Sound namespace.
    PlayRepeatingWave(std::wstring(L"\\windows\\ring.wav"), 3000);
}
catch(std::runtime_error ex)
{
    // Unable to play the ringtone, does the file exist?
}
```

Use **StopRepeatingWave** to stop playing the tone. This is shown in this example:

Example 22.20 Stopping A Repeating Tone In C++

```
// The user responded, don't need to play the ring tone any more.
// From the PsionTeklogix::Sound namespace.
StopRepeatingWave();
```

Setting The Speaker Volume

Use **SetSpeakerVolume** to control the volume of the speaker on the Psion Teklogix computer. 0 (zero) is the minimum volume of the speaker on the Psion Teklogix computer, and 100 is the maximum volume of the speaker. The sound level is hardware dependent; on different Psion Teklogix computers the same setting gives different apparent sound levels. Zero volume does not turn the speaker off; it is the lowest sound level available.

The volume can take one of the following values: 0, 20, 40, 60, 80,100. Setting any other value results in rounding to the nearest valid value.

In order to set the speaker volume, there must be an active call when the volume is set.

Configuring The Microphone

At this time (April 2008), the only available microphone operation is muting.

Muting The Microphone

Use **SetMicrophoneMute** to mute the microphone on the Psion Teklogix computer. When mute is enabled, the other parties cannot hear you even if the call is in the active state: You can still hear the other parties.

Use **GetMicrophoneMute** to check the mote status of the microphone.

In order to mute the microphone, there must be an active call when mute is set.

22.12.14 Voice Over WWAN Events

22.12.14.1 Voice Call Status Changed Event

A voice call status changed event is generated when one of the following changes for a voice call:

- The voice call state. See “Voice Service States” on page 217 for details.
- The voice call flag.

Voice Call Flags

The voice call flags define the type of voice call.

Incoming

This is an incoming call. Another party initiated this voice call.

Conference

This call is part of a multiparty call. See “Multiparty Calls (Conference Calls)” on page 226.

Last Event

The voice call is terminated—no more events will be generated for this call. See “Terminating Voice Calls” on page 227 for details.

Withheld Incoming Number

This is an incoming call where the caller withheld the phone number. See “Blocking Caller ID On Outgoing Calls” on page 228 for details.

22.12.14.2 Voice Call Manager Incoming Call Event

A voice call manager incoming call event is generated when a new voice call is received. Events are processed in the order that they are received. The application does not receive the next event until the handler returns from the previous one. For this reason, it is recommended that there is a lightweight handler that returns quickly.

22.12.15 Voice Over WWAN API Elements

C++: The phone book on all Psion Teklogix computers is controlled using the **VoiceCall** class and the **VoiceCallManager** class in the **PsionTeklogix::WWAN** namespace.

Java: Not available.

.NET: WWAN on all Psion Teklogix computers is controlled using the **VoiceCall** class and the **VoiceCallManager** class in the **PsionTeklogix.WWAN** namespace.

22.13 Phone Books

The Mobile Devices SDK allows an application to access and manipulate the phone books built into a SIM card or WWAN modem.

Each phone book entry consists of one name and one phone number. The maximum number of bytes for a name and for a phone number is determined by the network provider. The number of characters that can be stored depends on the alphabet used for the entry. 16-bit alphabets allow fewer characters to be stored in the available space than do 7-bit, and 8-bit alphabets. A name entry that is too long for the allocated space is automatically truncated. The number can only contain the following characters: 0 (zero) to 9, *, #, and +. Hyphens and brackets are automatically stripped from numbers before they are stored.

Reading Phone Book Entries

Use the following steps to read the entries from a phone book:

1. Get a list of all phone books—on both SIM cards and the modem—using **GetPhonebooks**.
2. Get the entries in a phone book using **GetPhonebookEntries**.

Example 22.21 Adding Contacts From A SIM Card To A Combo Box in C++

```
if (!InitWwanVoiceApi())
{
    return false;
}

Phonebook[] phonebooks = Phonebook.GetPhonebooks();
foreach (Phonebook phonebook in phonebooks)
{
    if (phonebook.StoredOnSim)
    {
        PhonebookEntry[] entries = phonebook.GetPhonebookEntries();

        foreach (PhonebookEntry entry in entries)
        {

            // Some phonebook entries are duplicates, or
            // they have zero length names. This example filters these out.
            if (
                (!number_combo.Items.Contains(entry.GetName())) &&
                (entry.GetName().Length > 0)
               )
```

Example 22.21 Adding Contacts From A SIM Card To A Combo Box in C++

```
            {
                number_combo.Items.Add(entry.GetName());
            }
        }
    }
}


if (number_combo.Items.Count > 0)
{
    number_combo.SelectedIndex = 0;
}
```

Writing Phone Book Entries

To create a new phone book entry:

- Use the **AddPhonebookEntry** method for the selected **Phonebook** object. A **PhonebookEntry** object representing the new phone book entry is returned.

Volatile

A phone book is *volatile* if its content can change at any time. Modems change this type of phone book, as well as the Mobile Devices SDK. *Dialled Calls*, *Received Calls*, and *Missed Calls* phone books are volatile; however, *Regular*, *Service Numbers*, *Own Numbers*, and *Voicemail Access Number* phone books are never volatile. If you want to make sure that you never miss a change in a volatile phone book, then you must re-read them whenever a call is disconnected.

Writable/Modifiable

Depending on the types of entries stored in a phone book, it may be writable or it may be read-only. See "Phone Book Types" on page 232 for more information.

22.13.1 Data Coding Schemes For Phone Book Entries

The data coding scheme for a phone book entry determines how many characters can be stored in the entry. For example, you can store approximately twice as many characters using an 8-bit alphabet as you can with a 16-bit alphabet.

A phone book can contain entries that use different alphabets.

The following alphabets can be used to store phone book entries:

- SMS Default
- SMS GSM 7-bit
- SMS ASCII 8-bit
- SMS Unicode 16-bit
- SMS Binary 8-bit
- SMS GSM PDU

Unicode entries must be stored in a 16-bit alphabet.

SIM cards can only store entries in SMS GSM 7-bit and SMS Unicode 16-bit.

The SMS Default uses the driver default algorithm for selecting the alphabet for an entry. The driver examines the characters used in the name, and the driver selects the alphabet that stores the maximum number of characters in the space available for the entry.

22.13.2 Phone Book Types

A SIM card or WWAN modem usually has several phone books stored on it, each containing a specific type of phone number. Each SIM or modem has a subset of the possible phone books. The following phone books types are identified and processed by the Mobile Devices SDK:

Phone Book	Type	Description
Regular	Writable	Features depend on the SIM or the modem.
Services	Not writable	Network provider numbers such as numbers for technical assistance.
Mailboxes	Writable	Numbers for contacting voice mail. Overwriting these numbers is not advisable.
Own Numbers	Writable	The phone number for the Psion Teklogix computer. This is for display purposes only—the usable number is stored on the network.
Dialed Calls	Not writable, volatile	Automatically updated by the modem.
Received Calls	Not writable, volatile	Automatically updated by the modem.
Missed Calls	Not writable, volatile	Automatically updated by the modem.
White List	Read only	For future use.
Black List	Read only	For future use.
Emergency	Read only	Built-in emergency numbers.

22.13.3 Phone Book API Elements

C++: The phone book on all Psion Teklogix computers is controlled using the **Phonebook** class and the **PhoneBookEntry** class in the **PsionTeklogix::WWAN** namespace.

Java: Not available.

.NET: WWAN on all Psion Teklogix computers is controlled using the **Phonebook** class and the **PhoneBookEntry** class in the **PsionTeklogix.WWAN** namespace.

22.14 Resource Materials

European Telecommunications Standards Institute (ETSI): Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Alphabets and language-specific information (3GPP TS 23.038 version 6.1.0 Release 6).

European Telecommunications Standards Institute (ETSI): Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Short Message Service (SMS) (3GPP TS 23.040 version 6.5.0 Release 6).

European Telecommunications Standards Institute (ETSI): Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); AT command set for 3G User Equipment (UE), 3GPP TS 27.007 version 5.4.0 Release 5, ETSI, 09/2003; Available at webapp.etsi.org/key/queryform.asp

ETSI reports are available at www.etsi.org.

D. Živadinovic, M. Winkler, "Des Surfers Bastelstunde" in c't 7/2001, page 228 (in German).

APPENDIX A

RESOURCES

A.1 Manuals and URLs

Psion Teklogix USB Setup Utility

Psion Teklogix Teknet:
https://teknet.psionteklogix.com/ptxCMS/Teknet.aspx?s=us&p=SoftwareDownloads

Microsoft Windows CE & Windows Mobile

Microsoft Inc.:
http://msdn.microsoft.com/en-us/library/ms376734.aspx

Psion Teklogix SDKs

Psion Teklogix. 2008. Psion Teklogix Mobile Devices SDK C++ API Reference Manual (Part number 8200104).

Psion Teklogix. 2008. Psion Teklogix Mobile Devices SDK .NET API Reference Manual (Part number 8200105).

Psion Teklogix. 2008. Psion Teklogix Mobile Devices SDK Java API Reference Manual (Part number 8200106).

Psion Teklogix. 2007. Psion Teklogix Imaging Services SDK Developers Guide (Part number 8100153)

7535 Hand-held Computer

Psion Teklogix. 2007. Psion Teklogix 7535G2 Hand-Held Computer User Manual (Part number 8100075)

7530 Hand-held Computer

Psion Teklogix. 2006. 7530G2 Hand-Held Computer User Manual (Part number 8100081)

8515 Vehicle-mount Computer

Psion Teklogix. 2007. Psion Teklogix 8515 Vehicle-Mount Computer User Manual (Part number 8100132)

8525/8530 Vehicle-mount Computer

Psion Teklogix. 2007. Psion Teklogix 8525/8530 G2 Vehicle-Mount Computer User Manual (Part number 8100083)

Ikôn Hand-held Rugged PDA

Psion Teklogix. 2009. Ikôn Rugged PDA (Windows CE 5.0) User Manual (Part number 8100147)

Psion Teklogix. 2009. Ikôn Rugged PDA (Windows Mobile 6.0) User Manual (Part number 8100149)

Psion Teklogix. 2009. Ikôn Rugged PDA (Windows Mobile 6.1) User Manual (Part number 8100181)

NEO Hand-held Computer

Psion Teklogix. 2008. NEO Hand-Help Computer User Manual (Part number 8100157)

WORKABOUT PRO G2

Psion Teklogix. 2007. WORKABOUT PRO Hardware Development Kit User Manual (Part number 8100057)

Psion Teklogix. 2007. WORKABOUT PRO G2 Windows Embedded CE 5.0 User Manual (Part number 8100140).

Psion Teklogix. 2007. WORKABOUT PRO G2 Windows Mobile 6 User Manual (Part number 8100144).

Psion Teklogix. 2009. WORKABOUT PRO G2 Windows Mobile 6.1 User Manual (Part number 8100182).

Development Environment / Compilers

Microsoft Visual Studio 2005/2008

http://msdn.microsoft.com/en-us/vstudio/default.aspx

Java Utilities

J2SDK:
http://java.sun.com/products/archive/

IBM J9 JVM (WebSphere Everyplace Micro Environment):
http://www-01.ibm.com/software/wireless/weme/.

Scanners

Symbol Technologies Inc. MiniScan scanner manuals:
www.symbol.com/services/manuals/scanner/miniscan.html

PSC Inc. (PowerScan scanners): http://www.psc.com

Symbol Decoded Scanner Configuration Codes

Symbol Technologies Inc. 2004. MiniScan MS XX04 Series Integration Guide (Part number 72-67134-02): Chapter 11 Parameter Menus.

Imaging Scanners

Symagery: http://www.symagery.com

BSQARE JEM-CE JVM

Psion Teklogix. 2005. BSQUARE JEM-CE Java Virtual Machine For Psion Teklogix Computers (Part number 8100054).

Windows CE programming

Microsoft. Programming Microsoft Windows CE .NET. Douglas Boling.

Wireless Wide-area Networking

European Telecommunications Standards Institute (ETSI): Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Alphabets and language-specific information (3GPP TS 23.038 version 6.1.0 Release 6).

European Telecommunications Standards Institute (ETSI): Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Short Message Service (SMS) (3GPP TS 23.040 version 6.5.0 Release 6).

European Telecommunications Standards Institute (ETSI): Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); AT command set for 3G User Equipment (UE), 3GPP TS 27.007 version 5.4.0 Release 5, ETSI, 09/2003; Available at http://webapp.etsi.org/key/queryform.asp.

ETSI reports are available at http://www.etsi.org.

Summit Radios

Summit Data Communications Inc. Summit User's Guide; Available at: http://www.summitdatacom.com/documentation.htm.

Help File Registration

The Helpware Group. H2Reg available at: http://helpware.net http://www.summitdatacom.com/documentation.htm.

APPENDIX B

REGISTRY KEYS

B.1 WORKABOUT PRO Registry Keys

B.1.1 Registry Settings For Controlling VGA

The WORKABOUT PRO VGA settings are located in the following registry subkey:

HKEY_LOCAL_MACHINE\Drivers\Display\NVDDI

Registry Values

QVGA mode (REG_DWORD)

00 = VGA

non-zero = QVGA

Note: This value is only available for the WORKABOUT PRO G2.

B.1.2 Registry Settings For Controlling Scanner Power

This registry key is in effect as long as there is no scanner or imager configured. It applies to the following:

- WORKABOUT PRO—5 volt scanner only.
- WORKABOUT PRO G2—3.3 volt, and 5 volt, scanners.

The scanner power settings are located in the following registry subkey:

HKEY_LOCAL_MACHINE\Drivers\PsionTeklogix\Scanner

Registry Values

QVGA mode (REG_DWORD)

00 = no effect (default)

non-zero = scanner power is always on while the computer is powered up.

Note: This value is only available for the WORKABOUT PRO G2.

B.1.3 Registry Settings For Serial Ports

The WORKABOUT PRO serial settings are located in the following registry subkey:

HKEY_LOCAL_MACHINE\Drivers\PsionTeklogix\Serial

Note: These setting are specific to the platform and should be modified with care.

Registry Values

HardwareFlowBaud (REG_DWORD)

Setting this value enables hardware RTS/CTS on the FFUART of the WORKABOUT PRO G2.

non-zero = the minimum baud rate where hardware flow control is enabled. The maximum rate is 921.6 k baud.

00 = enable all baud rates.

Note: This value is only available for the WORKABOUT PRO G2.

Index

Setting this value moves the COM port to a different location.

B.1.4 Registry Settings For Psion Teklogix Device Drivers

The WORKABOUT PRO device driver settings are located in the following registry subkey:

HKEY_LOCAL_MACHINE\Drivers\PsionTeklogix\Expansion Slot

Registry Values

FFUART (REG_DWORD)

Setting this value enables hardware RTS/STS on the FFUART of the WORKABOUT PRO G2. If the key is absent, the FFUART COM1: port is not enabled.

1 = enables the full-function UART (FFUART) using the standard serial driver as COM1:.

0 = the FFUART COM1: port will not be enabled. (Default)

PCMCIA (REG_DWORD)

1 = enables the PCMCIA socket on the 100-pin connector—socket 1. The PCMCIA pins become unavailable for other uses. The pins defined for this slot have predefined meanings and must be adhered to in the hardware design.

0 = the PCMCIA socket is not be enabled. (Default)

USB (REG_DWORD)

1 = enables the USB hub and the 100-pin connector USB power control.

0 = neither the USB hub nor the USB power control signal will be enabled for the100-pin connector. (Default)

Note: If the USB value is absent or has a value of 0, the expansion module USB power control can still be controlled by the HDK API library.

B.1.5 Registry Settings For Non-Psion Teklogix Device Drivers

All non-Psion Teklogix device drivers must have a registry entry. These device driver entries are formatted as follows:

HKEY_LOCAL_MACHINE\Drivers\PsionTeklogix\Expansion Slot*EEPROM*

Where: *EEPROM* is the name of the device driver. This is the contents of the **Manufacturer/Model** field in the expansion module EEPROM. For details see the *WORKABOUT PRO HDK Developers Manual*.

B.1.5.1 Loading Non-Psion Teklogix Drivers

At system startup the following process is used to load non-Psion Teklogix device drivers:

1. The contents of the **Manufacturer/Model** field in the expansion module EEPROM are appended to the registry key
   HKEY_LOCAL_MACHINE\Drivers\PsionTeklogix\Expansion Slot

For example, if the **Manufacture/Model** field contains:

ACME gizmo

then the following driver is loaded:

HKEY_LOCAL_MACHINE\Drivers\PsionTeklogix\Expansion Slot\ACME gizmo

2. The **DriverActivate()** function uses this registry key to activate the driver.

No driver is loaded if:

- The **Manufacturer/Model** field is not valid, or it is empty.
- The derived registry key does not exist.

Note: If multiple device drivers are required for the same device, subkeys are defined. Only the first driver is automatically activated. The application must load and activate all additional device drivers.

INDEX

.NET Namespaces *23*

A

Accepted Start Character *99*
Access flags *212*
  Set PIN *212*
  Set provisioned contexts *212*
  Set register state *212*
  Set suppress call notifications *212*
Access point name *211*
Accidental key presses *63*
AdaptiveWindowing *96*
Adaptor event *67*
Addendum *103, 104, 106, 108, 109, 110, 111, 113*
Addendum Add-on 2 *105, 107, 110, 112, 113*
Addendum Add-on 5 *105, 107, 110, 112, 113*
Addendum Required *105, 107, 110, 112*
Addendum Security *113*
Addendum Separator *105, 107, 110, 112*
AIAG Strip *97*
Aim Duration *92*
Aiming Mode *92*
Ambient light sensor *29*
API Reference Manuals *9*
APN *211*
Append *100*
Application lock-down mode *63*
AsyncMAC miniport *75*
Australia Post Address Code *87*
AutoExposure *95, 96*
Autorestore, Total Recall *172*
Aztec Code *88*
Aztec Runes *144*

B

Backlight *29*
  Configuration *31*
  Threshold *29*
  Timeout *29*
Backup, Total Recall *172*
Backup profile, Total Recall *171*
Bad Scan Beep *91*
Bar Code Field Settings *96*
Bar codes
  Initiating scan *150*
  Reading *150*
Barcodes Must Decode *95*
Battery
  Lifetime *39*
  Power management *37*
  Remaining charge *39*
  Threshold *38*
Beepers *161*
Berkeley Software Distribution interface *79*
Bi-directional Redundancy *93*
Blocking calls *228*
Bluetooth *155*
Bookland Enabled *113*
Broadband wireless *193*
BSD *79*

C

C++ Namespaces *13*
Call barring *216*
Call blocking *228*
Caller ID *216*
Call forwarding *216, 227*
Call hold *216*
Calling line identification *216*
Call waiting *216, 225*
Captures Per HW Trigger *96*
Card slots *39, 155*
CC-AB Enabled *133, 134*
CC-C Enabled *133, 134*
CDMA *196, 197*
Center Barcode Only *95*
Challenge Handshake Authentication Protocol *75*
CHAP *75*
Charge remaining, battery *39*
Check Char *100, 116, 125*
Check Digit(s) *105, 107, 110, 112, 119, 120, 122, 123, 145*
Check Digit Algorithm *123*
Check Digits Required *120*
Check Digit Verification *98, 99, 102, 116, 119, 121, 122, 124, 127*
Click Data *91*
Click Time *91*
Cloning *171*
Closing WWAN *205*
CLSI Editing *114*
CLSI Library System *116*
Codabar *87*
Codabar settings *114*
Codablock A Enabled *140*
Codablock F Enabled *140*
Codablock settings *139*
Code 11 *87*
Code 11 settings *119*
Code 128 *87*
Code 128 Emulation *137, 138*
Code 128 settings *101*
Code 16K settings *138*
Code 32 Prefix *98*
Code 39 *87*
Code 39 settings *97*
Code 49 settings *139*
Code 93 *87*
Code 93 settings *116*
Colours, display number of *47*

Command insertion *62*
Command key *62*
Command syntax *5*
COM ports
  Ikon *157*
  NEO *157*
  WORKABOUT PRO G2 *155*
  7530/7535 *156*
  8515 *156*
  8525/8530 *156*
Composite Codes *88*
Composite settings *132*
Concatenation *116*
Conference calls *216, 226*
Configuring scanner properties *90*
Configuring WLAN *187*
Connecting WWAN *210*
Connection Manager *76, 209*
ConstantIllumination *96*
ContinuousScanMode *93, 95*
Control Panel *213*
Conventions, text *4*
Convert to Code 32 *98*
Convert to EAN 13 *107, 109, 121*
Convert to UPC-A *110, 112*
Convert UPCE to UPCA *111*
Convert UPCE1 to UPCA *111*

D

DbGprs.csv *209*
Decoded scanners *85, 86, 87*
Decode Performance *98, 101*
Decode Performance Level *98, 101*
DecoderTimeout *96*
DefaultDevice *96*
Delete Character Set ECIs *94*
Dial-up, WWAN *196*
  CDMA *197*
  GSM *196*
  iDEN *197*
  UMTS *197*
Discrete 2 of 5 *87*
Discrete 2 of 5 settings *126*
Display
  Backlight *29*
  Information *47*
  Type *47*
Docking station *67*
  Events *67*
Dot Time *92, 95*
Double-click *181*
  Events *184*
  Settings *90*
DTMF *228*
Dual-tone multifrequency (DTMF) *228*

E

EAN-UCC Composite Enabled *134*
EAN-UCC Emulation *134*
EAN 128 *102*
EAN 128 Identifier *102*
EAN 13 *87*
EAN 13 Settings *103*
EAN 8 *87*
EAN 8 Settings *105*
EAN8 Zero Extend *106*
ECI Decoder *94*
EEPROM *71*
Emulators *9*
Enable Bookland *103*
Enable Plessey *124, 125*
Encryption *188*
Err Accept *97*
Events
  Double-click *184*
  Power state *38*
  Scan complete *152*
  Scan failed *152*
  Scanner *152*
  Serial port *157*
  Single-click *184*
  Trigger *183*
Expand *112*
Explicit call transfer *216*
Exp to UPC A *110*
Extended Coupon Code *114*
External scanners *86*

F

Factory Defaults on Reboot *96*
FastConverge *95, 96*
FFUART
  Enabling *B-2*
Field Size *97*
FNC1 Conversion *102*
Format *129*
Forwarding calls *227*
Friendly name *181*
Full ASCII *97, 99, 100*
Full ASCII Conversion *98, 99*
Funk Software, see Odyssey Access Client

G

Global bar code options
  Configuring *92*
Good Scan Beep *91*
GSM *196*
GSM power driver *198*
GS1-128 *102, 134*
GS1-128/GS1 US *101*
GS1-128 Identifier *102*
GTIN *113*
GTIN Compliant *102*

H

Held calls, WWAN *226*
HHP IT4080 *86*

I

IATA 2 of 5 *87*

IATA 2 of 5 settings *128*
iDEN *196, 197*
Ikon
  COM ports *157*
Imaging scanners *85, 86, 87*
Incall supplementary services *216*
Inc Country *103*
Include Check *99, 121*
Include Chk *97, 103, 106, 108, 110, 119, 120, 123, 126, 128*
Include Sym *101*
Inc Num Sys *108, 110*
Indicators, LED *51*
Interface event *67*
Interleaved 2 of 5 *87*
Interleaved 2 of 5 settings *120*
Intermec EV15 *86*
Intermec E1022 *86*
Internal scanners *86*
Internet connection
  WWAN *205*
Inverse Video Mode *141, 142*
IrDA *155*
ISBN Conversion *105*
ISBN Translate *105*
ISBT Concatenation *103*
ISBT Concatenation Pair *102*
ISBT Concatenation Transmission *102*
ISBT 128 *101, 102*
ItfChk *120, 126, 128*

J

Japanese Postal Code *87*
Java Packages *19*
Juniper Networks *187*

K

Key
  Command *62*
  Insertion *62*
  Modifier *57, 62*
  Presses, accidental *63*
Keyboard
  Backlight *29*
  disabling *56*
  Non-chorded *57*
  remapping *56*
  scan codes *58*
  supported *55*
  wedge *63*
Keyguard *38*
Korean Postal Code *87*

L

Laser On Time *92*
LED
  Colours *51*
  Illumination patterns *52*
  Indicators *51*
  Pulses *52*
Length L1 *98, 99, 102, 114, 116, 117, 118, 119, 121, 122, 123, 124, 125, 127, 129*
Length L2 *98, 114, 116, 117, 121, 122, 123, 127*
Length L3 *116, 122, 127*
Length Max *100, 103, 116, 118, 120, 122, 125, 126, 128, 129, 130, 132, 134, 137, 138, 139, 140, 141, 142, 143, 144, 149*
Length Min *100, 103, 116, 118, 120, 122, 125, 126, 128, 129, 130, 132, 134, 137, 138, 139, 140, 141, 142, 143, 144, 149*
Length Mode *116, 122, 127*
Lifetime, battery *39*
Linear Code Type Security Levels *93*
Linear Decode *113*
Linear Transmission only *134, 135*
Lock-down mode *63*
Locking permanent storage *71*

M

Matrix 2 of 5 settings *125*
MaxCapturesPerTrigger *96*
MaxGain *95, 96*
MaxIllumination *95, 96*
Maximum Length *100, 103, 116, 118, 120, 122, 125, 126, 128, 129, 130, 132, 134, 137, 138, 139, 140, 141, 142, 143, 144, 149*
Maximum Size *97*
MaxIntegration *95, 96*
MaxNumberBarcodes *95*
Micro PDF-417 settings *137*
Microsoft Challenge Handshake Authentication Protocol *75*
Minimum Cancel Time *93*
Minimum Length *99, 100, 102, 103, 116, 118, 119, 120, 122, 124, 125, 126, 128, 129, 130, 132, 134, 137, 138, 139, 140, 141, 142, 143, 144, 149*
Minimum Size *97*
MinNumberBarcodes *95*
MinScanDuration *96*
Mod Chk Base *97, 120, 126, 128*
Modems, WWAN *193, 198*
Modifier key *57, 62*
MOD 10 Check *120, 126, 128*
MSCHAP *75*
MSI Plessey *87*
MSI Plessey settings *123*
Multiparty
  Calls *226*
  Service *216*
Multiplexing, WWAN *194*

N

Namespaces, .NET *23*
Namespaces, C++ *13*
Narrowband radios *79*
NDIS *75*
NEO
  COM ports *157*
Network adapter, WLAN *189*
Network Driver Interface Specification *75*

Non-chorded keyboard *57*
Non-decoded scanners *85, 86, 87*
NOTIS Editing *114*
Number of colours, display *47*
Number System *110, 112*
Num Chk Digits *119*

O

Odyssey Access Client *187*
One Chk Digits *123*
OnlyDecodeCenter *95*
Output *130*

P

Packages, Java *19*
Packet data *197*
Packet data status *210*
PAP *75*
Parameter Scanning *93*
Password Authentication Protocol *75*
PCMCIA
  Enabling socket *B-2*
PDF417 settings *135*
Peripherals
  Events *67*
  Power modes *43*
Permanent storage *71*
  Locking *71*
Pharmaceutical *100*
Phone books *230*
Pistol grip trigger *181*
PLANET *88*
Plessey Length L1 *124*
Plessey Length Max *125*
Plessey Length Min *125*
Plessey Transmit Check Digit *124*
Point-to-Point Protocol *75*
Point-to-Point-Tunneling Protocol *75*
Port replicator *155*
Posicode *132*
PosiCode settings *132*
Postal
  Australian settings *146*
  Canadian settings *146*
  Japanese settings *147, 148*
  Korean settings *149*
  PlaNET settings *144*
  PostNET settings *145*
  Royal settings *150*
POSTNET *88*
Power
  Configuration *42*
  Control Panel *41*
  Modes, peripherals *43*
  State events *38*
Power Mode *93*
Power-on keyguard *38*
PPP *75*
PPTP *75*
Prefix *94*
Prefix Char *97*
Profile, Total Recall
  Backup *171*
  Content *171*
  Restoring *172*
  Storage locations *172*
  Types *171*
Proxy, remote sockets *79*

R

Radio signal strength *206*
RAS *75, 206*
Raster Expansion Rate *93, 94*
Raster Height *93, 94*
Reading Bar Codes *150*
Reading Range *99, 102, 122*
Received Signal Strength Indicator *188, 213*
Rectangular *141*
Registry
  Enabling FFUART through *B-2*
  Enabling PCMCIA socket through *B-2*
Remaining charge, battery *39*
Remapping, keyboard *56*
Remote Access Service *75, 206*
Remote sockets proxy *79*
Reset key *38*
Restore, Total Recall *172*
Restoring profiles, Total Recall *172*
Result Time *91*
Roaming WWAN *210*
Royal Mail Code *88*
RSS *88*
RSS Code settings *130*
RSS Expanded Enabled *130, 131, 132*
RSS Expanded Length Max *132*
RSS Expanded Length Min *132*
RSSI *188, 213*
RSS Limited Enabled *130, 131, 132*
RSS-14 Enabled *130, 131*
Runes Enabled *144*

S

Scan Angle *93*
SCAN button *181*
Scan codes, keyboard *55, 58*
Scan complete event *152*
Scan Data Transmission Format *94*
Scan Failed Beep *91*
Scan failed event *152*
Scan Good Beep *91*
Scan Indicator *91*
Scan Log File *91*
Scanner connections *85*
Scanner events *152*
Scanners *85*
  Configuring properties *90*
  Decoded *85, 86, 87*
  External *86*
  Imaging *85, 86, 87*
  Internal *86*

Non-decoded *85, 86, 87*
Unidentified *86*
Scanner setting index *89*
Scanner setting key *89*
Scanner setting name *89*
Scanning, bar codes *150*
Scanning Mode *93, 94*
Scan Result *91*
Scan Result Time *91*
Screen size *47*
SCU *187*
Security *92*
Security, Windows *177*
Security Level *113, 135*
Serial port, virtual *194*
Serial port events *157*
Set PIN access flag *212*
Set provisioned contexts access flag *212*
Set register state access flag *212*
Set suppress call notifications access flag *212*
Shell, Windows *177*
Short Code *92*
Short Message Service *215*
SIM card *200*
Single-click events *184*
SMS *200, 215*
Soft Scan Timeout *91*
Speakers *161*
Standard 2 of 5 Format *127*
Start Stop Char *100, 116*
StartStopChars *115*
Start Stop Transmission *99, 116*
Storage
Permanent *71*
Strip Leading *97*
Strip Trailing *97*
Suffix Char *97*
Suffix 1 *94*
Suffix 2 *94*
Summit Client Utility *187*
Summit mode *189*
Summit radios *189*
Suplementary services
Conference calls *216*
Supplemental Redundancy *113*
Supplementals *113*
Supplementary services
Call barring *216*
Caller ID *216*
Call forwarding *216*
Call hold *216*
Calling line identification *216*
Call waiting *216*
Explicit call transfer *216*
Incall *216*
Multiparty service *216*
WWAN *216*
Supported keyboards *55*
Suspend Timeout *37*
Symagery IL6303 *86*
Symagery SX4000 (ST) *86*
Symagery SX4000 (UHD) *86*
Symagery SX4000 (ULR) *86*
Symagery SX5003 HD *86*
Symagery SX5303 (ST) *86*
Symagery SX5303 (UHD) *86*
Symagery SX5303 (ULR) *86*
Symagery SX5303 IL *86*
Symagery SX5393 *86*
Symagery SX5400 *86*
Symbologies *87*
Symbol 1200 ALR *86*
Symbol 1200 HP *86*
Symbol 1200 LR *86*
Symbol 1200 WA *86*
Symbol 1223 *87*
Symbol 1223 ALR *86*
Symbol 1223 HP *86*
Symbol 1223 LR *86*
Symbol 1223 WA *86*
Symbol 1224 *87*
Symbol 1224 HP *86*
Symbol 1524 *87*
Symbol 1524 ER *86*
Symbol 2223 *86, 87*
Symbol 923 *87*
Symbol 923 HP *86*
Symbol 955 *86*
Syntax, command *5*

T

TekImager Enabled *95*
Telepen settings *129*
Terminating calls *227*
Tether port event *67*
Tether ports *68*
Text conventions *4*
Threshold, battery *38*
Time Delay To Low Power *93*
Timeout, backlight *29*
TLC-39 Enabled *133*
TLC-39 settings *134*
Total Recall *171*
Autorestore *172*
Backup profile *171, 172*
Profile content *171*
Profile storage locations *172*
Profile types *171*
Restore profile *172*
Touchscreen *47*
Transmit as EAN-13 *107, 109*
Transmit as UPC-A *112*
Transmit Check Digit *98, 99, 105, 107, 109, 112, 116, 119, 121, 122, 123, 124, 127*
Transmit Code ID Character *94, 97, 101, 103, 106, 108, 110, 115, 117, 119, 120, 123, 126, 128*
Transmit Number System *109, 110, 112*
Transmit power, WLAN *188*
Trigger
Associations *181*

Consumers *181*
Control flags *182*
Double-click *181*
Events *183*
Pistol grip *181*
SCAN button *181*
Source ID *181*
Sources *181*
Trioptic *98*
Trioptic Code Settings *100*

U

UCC EAN 128 *101*
UMTS *196, 197*
Unidentified scanners *86*
UPC/EAN *87*
UPC/EAN Shared Settings *113*
UPC A *87*
UPC-A Check Digit *108*
UPC-A EAN-13 Extended Coupon Code *114*
UPC and EAN composite message decoding *134*
UPCA Preamble *108*
UPC A settings *108*
UPCA Transmit Check Digit *108*
UPC E *87*
UPC-E Check Digit *111*
UPCE Enabled *111*
UPCE Preamble *111*
UPC E settings *110*
UPCE Transmit Check Digit *111*
UPC-E1 Check Digit *111*
UPCE1 Enabled *111, 112*
UPCE1 Preamble *111*
UPCE1 Transmit Check Digit *111*
UPC Half Block Stitching *113*
USB *155*

V

Variant Mode *101*
Verify *92*
Virtual-key code *55*
Virtual private network *75*
Virtual serial port *194*
Configuration *194*
Voice calls
Audio *228*
Blocking calls *228*
Call waiting *225*
DTMF *228*
Forwarding calls, WWAN *227*
Held calls *226*
Multiparty calls *226*
Processing *225*
Receiving *224*
Terminating calls *227*
WWAN *217*
WWAN closing *220*
WWAN initialization *217*
WWAN making *220*
Voice call states *222*
Active *222*
Alerting *222, 223*
Busy *223*
Dialing *222*
Held *222*
Incoming *223*
No answer *223*
No carrier *223*
Normal termination *222, 223*
Supplementary accepted *223*
Waiting *223*
Voice service states *217*
Disabled *217*
Enabled *217*
Some enabled *217*
VPN *75*
Connecting *211*

W

WAN *75*
WAV files *161*
Wedge, keyboard *63*
WEP *188*
Key index *188*
Key length *188*
Wide area network *75*
WindowHeight *95*
Windows Connection Manager *209*
Windows RAS API *206*
Windows security *177*
Windows shell *177*
Windows Sockets 2 *79*
WindowWidth *95*
Wired Equivalent Privacy *188*
Key index *188*
Key length *188*
Wireless broadband *193*
Wireless LAN, see WLAN
Wireless wide-area networking, see WWAN
Wireless Zero Config *187, 189*
Wizards *25*
WLAN *187*
API libraries *188*
Configuring *187*
Network adapter *189*
Transmit power *188*
WORKABOUT PRO G2
COM ports *155*
WWAN *193*
Access flags *212*
Access point name (APN) *211*
APN *211*
Closing *205*
Connecting *210*
Control Panel *213*
Dial-up *196*
Driver initialization *201*
Driver status flags *199*
Bad SIM *200*
Failure *200*

Initialized *200*
No SIM *200*
Phone book ready *199*
Shutdown *199*
SMS store ready *200*
Forwarding calls *227*
Initialization *199, 202, 217*
Initialization status *199*
Initializing voice calls *217*
Internet connection *205*
Modems *193, 198*
Multiplexing *194*
Packet data *197*
Phone books *230*
PIN authentication *201*
Roaming *210*
RSSI *213*
Supplementary services *216*
Voice call *217, 220*
Audio *228*
Blocking calls *228*
Call waiting *225*
Conference calls *226*
DTMF *228*
Events *229*
Held calls *226*
Making *220*
Multiparty calls *226*
Processing *225*
Receiving *224*
Terminating calls *227*
Voice call states *222*
Active *222*
Alerting *222, 223*
Busy *223*
Dialing *222*
Held *222*
Incoming *223*
No answer *223*
No carrier *223*
Normal termination *222, 223*
Supplementary accepted *223*
Waiting *223*
WZC *187, 189*

2D Aztec settings *143*
2D Data Matrix *88*
2D Data Matrix settings *140*
2 Digit Addendum *105, 107, 110, 112*
2D MaxiCode *88*
2D MaxiCode settings *142*
2D Micro PDF417 *88*
2D PDF417 *88*
2D QR Code *88*
2D QR Code settings *141*
5 Digit Addendum *105, 107, 110, 112*
7530/7535
COM ports *156*
802.11
Encryption *188*
Remote Sockets *79*
WLAN *187*
8515
COM ports *156*
8525/8530
COM ports *156*